Arquitetura de soluções IoT

Desenvolva com Internet das Coisas para o mundo real

FERNANDO BORBA FERREIRA
RENATO MANZAN
WELLINGTON DURAES

Casa do Código | alura

Sumário

- ISBN
- Introdução
- 1. Soluções IoT
- 2. Profissionais em projetos de IoT
- 3. Dispositivos IoT
- 4. Redes e protocolos
- 5. Serviços na nuvem para IoT
- 6. Cenários de IoT
- 7. Desafios arquiteturais de uma solução IoT
- 8. Conclusão
- 9. Referências

ISBN

Impresso: 978-85-5519-321-7

Digital: 978-85-5519-320-0

Caso você deseje submeter alguma errata ou sugestão, acesse http://erratas.casadocodigo.com.br.

Introdução

Tudo ao nosso redor está em processo de evolução: agricultura, educação, logística, segurança, sistemas de transporte, fabricação de produtos, controle do meio ambiente e até mesmo o modo como cuidamos de nossa saúde. Boa parte da evolução de cada uma dessas áreas está relacionada ao impacto de objetos comuns passarem a estar conectados à internet, ou *the Internet of Things* (IoT) – a Internet das Coisas.

Internet das Coisas é a união de objetos (por exemplo, veículos, ferramentas, prédios, roupas, eletrodomésticos e acessórios do dia a dia) com softwares, sensores, atuadores e alguma forma de conectividade que permita a coleta e troca de dados com um sistema distribuído. Entenda que este sistema distribuído pode tanto ser um serviço rodando em um *data center* ou servidor local, como um serviço hospedado na nuvem.

A evolução das tecnologias relacionadas a IoT viabiliza cenários e soluções que há poucos anos eram disponíveis apenas a grandes empresas ou mediante grandes investimentos em infraestrutura, que estavam completamente fora de alcance e dos planos de muitas empresas.

Embora o termo Internet das Coisas seja relativamente recente, soluções que conectam dispositivos a *data centers* ou sistemas distribuídos já existem há algumas décadas em cenários como: caixas eletrônicos, telecomando em subestações de energia elétrica e monitoramento em tempo real da frota de ônibus de uma cidade. Todos esses projetos exigiram um grande investimento por parte das empresas, além do alto risco e do tempo considerável para a implementação de toda solução.

A expectativa é de que existam mais de 1 trilhão de dispositivos IoT em funcionamento até 2030 (LEA, 2020). A premissa de que apenas computadores e laptops poderiam conectar-se à internet é obsoleta. Dispositivos conectados estão em todo lugar e a tendência é que

este número apenas cresça. Alguns dos fatores que favorecem o crescimento do mercado de IoT são (KHVOYNITSKAYA, 2021):

- Queda no custo de produção de sensores;
- Queda no custo de coleta e armazenamento de dados devido a soluções na nuvem;
- Expansão do acesso à internet;
- Crescimento do poder computacional.

Ao acompanhar as previsões do mercado, sempre são apresentadas quantias estratosféricas que motivam as empresas a investirem mais no setor, o que proporciona a criação de novos empregos, novas tecnologias e oportunidades para todos.

A popularização de IoT e do movimento de *makers* fez as pessoas acreditarem que um dispositivo de hardware qualquer, como um *Arduino*, e um serviço online gratuito são suficientes para criação de soluções complexas. Diversas *startups* surgiram nos últimos anos mesclando hardware, software e internet para fornecer novas soluções IoT ao mercado.

NOTA

Maker é um termo que tem se popularizado nos últimos anos e que remete ao espírito de "faça você mesmo", mas com foco em tecnologia. *Makers* são pessoas com interesses diversos relacionados a eletrônica, robótica, impressão 3D, máquinas de Controle Numérico Computadorizado (CNC), combinadas ou não com marcenaria, metalurgia, costura e praticamente qualquer outra forma de arte.

Arduino é uma plataforma eletrônica de código aberto baseada em hardware e software fáceis de usar.

Embora seja verdade que a tecnologia tenha democratizado o acesso a recursos computacionais a um grande número de

pessoas, construir soluções IoT não é uma tarefa trivial. A criação de sistemas profissionais de IoT é complexa e bastante abrangente. Um sistema bem arquitetado exige domínio sobre diferentes áreas de conhecimento, desde sensores e considerações sobre consumo de energia até protocolos de comunicação e serviços providos por serviços na nuvem.

Objetivo

O objetivo deste livro é criar um manual de referência em IoT, considerando aspectos teóricos e práticos, para que pessoas desenvolvedoras de sistemas, pessoas arquitetas de solução e estudantes possam utilizá-lo no dia a dia.

Mais do que compartilhar técnicas sobre IoT, essa publicação visa descrever conceitos, desde dispositivos, sensores e meios de comunicação com serviços na nuvem, até possíveis formas de como endereçar desafios arquiteturais de soluções IoT.

Como este livro está organizado

O livro está organizado da seguinte forma: o capítulo *Soluções IoT* descreve as características de uma solução IoT, assim como cita suas motivações e seu impacto no mercado de tecnologia. O capítulo *Profissionais em projetos de IoT* discute os papéis desempenhados por diferentes tipos de profissionais na construção de soluções IoT. O capítulo *Dispositivos IoT* apresenta diferentes tipos de dispositivos, sensores e práticas comuns na manutenção destes dispositivos. O capítulo *Redes e protocolos* lista e detalha os tipos de redes e protocolos comumente utilizados na comunicação entre dispositivos em soluções IoT. O capítulo *Serviços na nuvem para IoT* descreve o uso de nuvem (assim como de sistemas distribuídos) em soluções IoT. O capítulo *Cenários de IoT* descreve soluções adotadas para diferentes desafios técnicos em IoT e o capítulo *Desafios arquiteturais de uma solução IoT* resume diferentes desafios presentes na construção de soluções IoT.

Para quem é este livro

- Profissionais de tecnologia que desejam aprofundar seus conhecimentos em IoT;
- Arquitetas e arquitetos de software que buscam aprender os desafios por trás da arquitetura de soluções IoT;
- Profissionais que possuem interesse em compreender suas responsabilidades no desenvolvimento de soluções IoT.

Prefácio, por Reginaldo Arakaki

Nesta obra, Fernando Borba Ferreira, Renato Manzan e Wellington Duraes alcançam, de maneira brilhante, a formalização de conhecimentos na concepção, arquitetura e construção de sistemas que incluem os atributos de *Internet of Things*.

Os assuntos aqui organizados interessam a pessoas envolvidas em sistemas de TI. Aplicações e soluções são amplamente discutidas para introduzir o quanto os eventos do mundo real, em seus mais variados contextos de aplicação, podem ser coletados, tratados e devolvidos para os diversos processos que envolvem pessoas e coisas. Com isso, os perfis de profissionais são discutidos no âmbito de IoT em diversos aspectos de maneira ampla, indicando como o conhecimento e experiências podem ser ampliadas na adoção da tecnologia. Sempre lembrando que soluções que envolvem IoT abrangem todas as áreas onde as plataformas digitais estão presentes: educação, transporte, comércio, moradia, saúde, finanças e segurança, além de estarem presentes, também, nas empresas, no poder público e no dia a dia das pessoas.

O(a) leitor(a) encontra neste livro uma discussão sobre a conexão do mundo real (analógico) com o mundo digital através do entendimento tanto dos dispositivos que coletam dados como também dos que atuam sobre os elementos do mundo real. Como

exemplo, plataformas IoT para ajudar pessoas com necessidades específicas de acompanhamento de taxas de glicose no sangue poderiam utilizar quais tipos de dispositivos? Como identificar as condições de normalidade, de exceções e avisar aos especialistas para tomar as devidas providências? Estes dispositivos coletam e atuam sobre os processos do mundo real e devem se comunicar com computadores e sistemas na retaguarda (*cloud*). Como prover esta conectividade de maneira precisa, segura e disponível? Além disso, estes eventos acontecem em volumes muito grandes e devem ser processados em tempo real, com requisitos de tempo e de acessos estabelecidos pelos processos aos quais atendem.

O livro discute cenários de aplicação de sistemas que envolvem as casas, a cidade, o carro e o corpo – todos conectados digitalmente por plataformas IoT, com suas camadas de complexidade e restrições de engenharia – e aponta diferenças entre os sistemas construídos de maneira profissional e os sistemas “montados” de maneira amadora.

O ponto alto desta obra é trazer para os leitores uma reflexão sobre os aspectos de engenharia que constituem um sistema robusto e flexível. São os desafios arquiteturais de uma solução IoT para embutir os requisitos não funcionais de um sistema digital, incluindo aspectos de usabilidade, desempenho, precisão, manutenção, eficiência no uso de energia e segurança.

Os sistemas IoT exigem cuidados no balanceamento das suas soluções e implementações, na medida em que combinam dispositivos sensores (muitas vezes delicados e suscetíveis a falhas, por estarem expostos a ambientes diversos, e conectados por tecnologias de comunicação de redes e protocolos como as redes locais e o 5G) a ambientes com grande poder computacional, como os disponibilizados em *Cloud Computing* e no Microsoft Azure. Aliadas ao conjunto muito grande de eventos (dados e ações com carimbos de tempo), muitas provocações são brilhantemente colocadas pelos autores.

Aproveitem muito desta obra. Ela representa um salto no conhecimento e aplicação de sistemas de TI, em um momento em que tecnologias como *digital twin*, inteligência artificial e *machine learning* baseiam-se em dados obtidos por IoT nas aplicações pessoais, domésticas ou industriais.

Parabéns pelo livro, Fernando, Renato e Wellington. Muito obrigado pelas contribuições na organização do conhecimento a ser disseminado com muito sucesso por esta obra.

Reginaldo Arakaki - *Professor doutor do Departamento de Engenharia de Computação e Sistemas Digitais da Escola Politécnica da Universidade de São Paulo*

Prefácio, por Olivier Bloch

Pense o seguinte: o termo "Internet das Coisas" (mais conhecido como IoT) contém as palavras "internet" e "coisas".

O uso da palavra "internet" é bastante óbvio: refere-se a conectividade, comunicação e ao acesso remoto para troca de dados e informações.

Já a adoção da palavra "coisas" não é tão clara. Não dizemos "dispositivos", "máquinas" ou "sensores". É realmente sobre qualquer "coisa" que possa se conectar à internet e compartilhar dados, receber comandos ou participar de um sistema conectado.

Os valores em se conectar "coisas" à internet são agora reconhecidos pela indústria de diferentes formas: seja no monitoramento e controle remoto de dispositivos, ou na ingestão e uso dos dados para melhoria de produtos, serviços, fluxos de trabalho, segurança etc., tenho certeza de que você mesmo poderá ver muitos valores nessa abordagem.

Mas é muito mais fácil falar do que fazer. Implementar uma solução IoT de verdade é complexo: desenvolver uma aplicação IoT do começo ao fim, dos sensores até a nuvem, exige um conhecimento muito amplo, desde fundamentos de eletrônica até a escalabilidade de recursos na nuvem. Por conta disso, a colaboração de diferentes profissionais com conhecimentos em diferentes domínios é necessária.

Neste livro, Fernando, Renato e Wellington – experientes conhecedores das engenharias de hardware e software – colocam seus chapéus de educadores para explicar IoT de ponta a ponta, de forma concreta e simples, para todos profissionais e entusiastas interessados em aventurar-se no mundo de IoT.

Nesta obra os autores descrevem do que soluções IoT são feitas e quem as constrói; o que as "coisas" são e como elas se conectam e se comunicam com a internet; e como serviços da nuvem são utilizados em soluções IoT – sempre embasando suas explicações com cenários e exemplos reais, além de detalhar os desafios arquiteturais presentes nessas soluções IoT.
Não importa se você é uma pessoa engenheira de software experiente ou alguém interessado em aprender o que é IoT: eu tenho certeza de que você irá apreciar esta concisa e completa obra para a construção de soluções.

Olivier Bloch - *Microsoft Principal PM Manager*

Sobre os autores

Fernando Borba Ferreira

Fernando Henrique Inocêncio Borba Ferreira atua como Software Engineer na Microsoft. Trabalhou no desenvolvimento do Microsoft Windows 10 e de produtos dentro das divisões do Azure IoT e Azure

Data. Atualmente, trabalha construindo soluções para a área de saúde dentro da divisão do Azure Health & Life Sciences. No passado, atuou como Premier Field Engineer em Modern Apps, dando suporte a sistemas críticos em todo território brasileiro. Fernando é bacharel e mestre em Sistemas de Informação. Com mais de 15 anos de experiência em desenvolvimento de software, ele também foi Microsoft MVP em Data Platform Development e Visual C#. Fernando passa seu tempo livre com sua família, livros, música, passeios ao ar livre e no auxílio a pesquisas acadêmicas.

Renato Manzan

Renato Manzan atua como Customer Success Account Manager na Microsoft. Em mais de 20 anos de experiência, já atuou como desenvolvedor, arquiteto, gerente de projetos e gerente de suporte. É bacharel em Ciências da Computação pela PUC-SP, especialista em Engenharia de Software pela Unicamp, mestre e doutor em Computação pela Escola Politécnica da USP. Também atua como professor e já ministrou as disciplinas de Arquitetura de Sistemas Distribuídos, Laboratório de Programação, Engenharia e Qualidade de Software, Modelagem Orientada a Objetos, Engenharia de Requisitos, Computação em Nuvem e de Arquitetura de Software. Atualmente, leciona no Curso de Especialização em Engenharia de Software na PUC-SP.

Wellington Duraes

Wellington Duraes atua como Program Manager no time do Azure IoT da Microsoft, responsável por SDKs que auxiliam dispositivos IoT a se conectarem com os serviços de IoT da Microsoft. É mestre em Engenharia de Computação e atua na área de tecnologia há mais de 20 anos, além de ter curso técnico em eletrônica, onde trilhou seus primeiros passos no incrível mundo do software embarcado. Além de trabalhar com IoT, em seu tempo livre prepara e ministra workshops relacionados a IoT para entusiastas e makers. Por diversão, cria projetos eletrônicos com diversos microcontroladores, desenha placas de circuito impresso, cria e

imprime modelos na impressora 3D e compartilha seu conhecimento em https://github.com/wduraes/Workshops.

Capítulo 1
Soluções IoT

1.1 Computação Ubíqua e sua evolução

Apesar de o termo IoT ser amplamente difundido nos últimos anos, a utilização de pequenos dispositivos conectados à internet não é recente. Até então, o uso de tais dispositivos era agrupado em um conceito mais amplo, chamado *Computação Ubíqua* (também conhecido como *Computação Pervasiva*).

A Computação Ubíqua (COULOURIS *et al.*, 2011) é uma área da Engenharia de Software onde são concentrados os avanços tecnológicos na integração de pequenos dispositivos em um ambiente físico com sistemas distribuídos. O principal objetivo da Computação Ubíqua é viabilizar o uso de computação e torná-la cada vez mais acessível, dada a portabilidade e a habilidade de dispositivos em se conectarem com redes de computadores em diferentes lugares.

O termo *ubíquo* é utilizado com o intuito de sugerir que pequenos dispositivos se tornaram tão comuns em nosso dia a dia que eles não serão mais notados como um avanço e, sim, como parte da nossa rotina diária. Não confunda Computação Ubíqua com Computação Móvel. Ambos os conceitos possuem muitas semelhanças, mas são diferentes: a Computação Móvel está relacionada à mobilidade de dispositivos computacionais, que permite que usuários tenham acesso a informações de qualquer lugar; a Computação Ubíqua destaca os benefícios que os usuários usufruem na utilização de dispositivos enquanto estão em um local fisicamente, seja dentro de casa, em um carro ou em uma loja.

1.2 Como a nuvem e novas tecnologias permitem novos cenários

Na categoria de nuvem, temos grandes provedores (Amazon, Google e Microsoft) disputando tanto em quantidade de recursos oferecidos quanto nos preços de seus serviços, o que faz com que o custo total de manter uma solução de IoT seja cada vez mais acessível.

Na categoria de dispositivos, destacam-se os microcontroladores (MCU, do inglês *microcontroller unit*), cada vez mais poderosos, mais baratos e consumindo cada vez menos energia, com baterias ainda melhores e células solares mais eficientes, além de diversas tecnologias disponíveis para conectá-los com o mundo.

A título de comparação, há pouco mais de uma década, um projeto para monitoração de frotas de ônibus de uma metrópole exigiria os seguintes itens da empresa que fosse implementá-lo:

- Desenvolvimento de um dispositivo para ser instalado nos veículos, com antena de GPS, conexão celular e software embarcado criado especificamente para esse dispositivo;
- Criação de um protocolo de comunicação que otimize a banda de telecomunicações (dado que a tarifação neste caso é feita por bytes trafegados);
- Criação de uma plataforma online para o consumo, armazenamento e entrega das mensagens enviadas pelos veículos aos demais sistemas para processamento;
- Criação de mecanismos que permitam a atualização segura do software dos dispositivos nos veículos quando necessário;
- Alocação de uma grande quantidade de servidores em um *data center* para suportar alta demanda no volume das mensagens recebidas, mantendo o tempo de resposta esperado durante horários de maior utilização, mesmo que os servidores fiquem subutilizados no restante do tempo.

É necessário dizer que o custo de uma solução dessa magnitude é alto e, muitas vezes, proibitivo para empresas menores – ou, nesse caso, cidades menores. Mas essa era a realidade para investir em uma solução de IoT há alguns anos atrás.

Atualmente, considerando os avanços da tecnologia na área de IoT, a mesma solução poderia se beneficiar de:

- Facilidade de uso de soluções de hardware modulares para GPS, conexão com celular e meios de comunicação de longa distância antes indisponíveis, como LoRa (*Long Range*), por exemplo;
- Utilização de protocolos otimizados para telemetria, como MQTT (*Message Queuing Telemetry Transport*), com baixo consumo de energia e otimização dos pacotes de dados trafegados;
- Utilização de SaaS (*Software as a Service* – em português, Software como Serviço), que oferece plataformas de ingestão de dados com alta capacidade, baixa latência e comunicação bidirecional. Essas plataformas são escaláveis e podem ter sua capacidade aumentada sob demanda, liberando recursos quando não são mais necessárias;
- Utilização de recursos da nuvem para ações de OTA (*Over-the-Air Update* – em português, atualização de software pelo ar), que permitem a atualização do software dos dispositivos sem a necessidade de se criar uma solução exclusivamente dedicada a isso;
- Utilização de recursos computacionais na nuvem, não sendo necessário construir ou manter um *data center*. Os grandes provedores de nuvem possuem diversos *data centers* espalhados pelo mundo, disponibilizando capacidade (virtualmente) ilimitada, onde usuários pagam apenas pelos recursos que consomem.

Com todos esses avanços, pequenas empresas podem criar soluções de IoT sem ter que investir uma quantidade significativa de

dinheiro e tempo de desenvolvimento, como era necessário no passado.

Como resultado dessa evolução, inúmeros cenários que antes não eram factíveis agora o são. As oportunidades para empresas e profissionais de IoT são maiores do que a mão de obra disponível, o que requer que novos profissionais se capacitem para criar essas soluções.

1.3 Tornando objetos comuns em dispositivos inteligentes

Um dos grandes benefícios de IoT está em transformar objetos triviais em dispositivos inteligentes. Objetos que até então não possuíam qualquer inteligência passam a fornecer dados telemétricos e a responder comandos externos a fim de aperfeiçoar seu funcionamento.

Um exemplo famoso no mercado ocorre com as máquinas de café expresso: tais máquinas são predecessoras à internet e por muitos anos funcionaram sem a necessidade de estarem online. Mas uma grande rede mundial de cafeterias passou a equipar suas máquinas de café expresso com sensores, conectando-as à internet para a transmissão de telemetria. E hoje, após anos de pesquisa, coleta e análise dos dados, a empresa já consegue:

- Identificar máquinas de café expresso que precisam de manutenção preventiva para que a qualidade do produto final não seja afetada;
- Quantificar a matéria-prima utilizada por dia na produção de cafés expressos e investir na otimização de recursos;

- Identificar cafeterias que, dado o volume de produção, estão superutilizando ou subutilizando suas máquinas de café e que possuem espaço para melhorias de processo e/ou distribuição de recursos.

Voltemos para um cenário mais simples, de algo que está presente em nossas casas e que faz parte de nosso dia a dia: a balança tradicional de banheiro. A balança apenas informa ao usuário o seu peso de forma transacional: o usuário sobe na balança, a balança calcula o seu peso e o exibe. A balança não mantém nenhum histórico. Caso o usuário queira acompanhar a evolução do seu peso, ele terá de fazer por si mesmo.

Usando IoT, uma balança agora pode prover mais valor e acompanhar a evolução do peso do usuário a cada transação: identificando o usuário que está se pesando; armazenando o peso do usuário temporariamente; conectando-se a um serviço que persista esta informação; recebendo a notificação de que essa informação foi persistida; apagando o registro temporário do peso do usuário, desconectando-se e voltando a seu estado de hibernação.

Com a adição de mecanismos de comunicação, a balança passa a fazer parte da categoria de dispositivos inteligentes. Ela proporciona valor agregado ao usuário, que pode, de posse do histórico de peso, verificar sua evolução ao longo do tempo, acompanhar seus objetivos e ainda cruzar essa informação com outros dados biométricos que sejam capturados por outros dispositivos ou até mesmo por essa balança - por exemplo, o percentual de gordura corporal.

Exercício

Esta atividade pode ser, ao mesmo tempo, desafiadora e geradora de boas ideias.

Olhe à sua volta, não importa onde você esteja neste momento. Procure por objetos do dia a dia que não são conectados à internet e se pergunte que tipo de benefício/valor esse objeto poderia fornecer a você se ele tivesse essa conexão com a internet.

Faça algumas iterações sobre as possibilidades e insista um pouco para cada objeto que você vê. Os resultados podem ser surpreendentes.

Devemos sempre nos questionar "que problema este dispositivo não endereça hoje e que, se estivesse conectado, ele poderia endereçar?"; "que cenários de negócio o dispositivo vai permitir ao se tornar conectado?"; "o custo-benefício de conectá-lo vale o esforço?". Quando a resposta a essas perguntas estiver clara, defender a execução do projeto ou justificar os motivos de não seguir com ele será uma tarefa mais simples e embasada em um processo claro.

1.4 Democratização da tecnologia

Enquanto criar soluções profissionais ainda requer um investimento considerável, pode-se dizer que a tecnologia atual democratizou o acesso e estudo de soluções que antes eram disponíveis apenas a empresas possuidoras de capital para tal empreendimento.

Em uma escala que vai do *maker* até o desenvolvedor de uma grande empresa, praticamente todos têm acesso à tecnologia e capacidade na nuvem, dados que – em sua maioria – são tarifados

por hora de uso. Ainda é necessário considerar que vários provedores oferecem serviços gratuitos: alguns deles por um período de tempo (*trial*) e outros sempre gratuitos.

Dentro das empresas, essa democratização se mostra quando departamentos fora da área de TI passam a consumir serviços de nuvem para realizar experimentos (provas de conceito) e até mesmo para manter pequenos sistemas independentes da equipe de TI. Esse conceito é conhecido como *shadow IT*. O baixo custo de manter pequenas soluções na nuvem contribui fortemente para isso.

Especificamente na área de IoT, além das capacidades específicas da nuvem, há ainda muitos dispositivos voltados para o desenvolvimento e testes. Por exemplo, kits de desenvolvimento (*dev kits*) como o Arduino Uno permitem o aprendizado e prototipação a baixo custo com dispositivos de hardware reais, ou seja, dispositivos físicos.

Como um(a) arquiteto(a) de soluções em IoT, é importante preparar-se para reaproveitar ou herdar protótipos de soluções em IoT criadas por áreas fora de TI, o que pode trazer diversas surpresas. O ponto crucial nesse cenário não é o reaproveitamento de dispositivos ou, ainda, o código legado que não segue boas práticas da engenharia de software, mas sim o entendimento do problema que se pretende resolver com o protótipo, que pode ser considerado uma especificação viva.

Ferramentas de *low code* têm se popularizado e auxiliado na criação de protótipos funcionais, pois ajudam pessoas a criarem soluções sem a necessidade de escrever código, apenas por meio de uma interface gráfica.

Ao invés de condenar e/ou proibir sua criação, as empresas deveriam incentivar o uso de ferramentas *low code*, dado que, ao experimentar/prototipar suas soluções, as áreas de negócio podem iterar e entender melhor seus próprios requisitos, poupando preciosos esforços dos times de tecnologia. Basta que as áreas de

negócio entendam que seus protótipos não passam de artefatos de teste de seus requisitos funcionais, não sendo adequados para utilização em produção. A implantação em produção demanda uma implementação condizente com o uso de práticas da engenharia de software.

1.5 Desafios de uma solução de IoT

As características das soluções de IoT trazem consigo desafios únicos no que diz respeito a seus dispositivos e à rede necessária para conectar esses dispositivos com a nuvem.

Mudanças no hardware

Imagine um cenário em que sua aplicação de comércio eletrônico está concluída, mas no último instante uma modificação para inserir um novo campo em um dos formulários web é necessária. Embora tal mudança possa ser relativamente complexa, se ela for cuidadosamente planejada, pode ser realizada com boas chances de sucesso.

Mudemos o cenário para uma aplicação IoT de monitoramento ambiental de uma cidade: neste exemplo, dispositivos foram implementados em conjunto com as centenas de relógios públicos da cidade, mas, no momento final dos testes, descobre-se que o sensor utilizado no dispositivo não é adequado para uso externo e apresenta uma variação inaceitável dos valores coletados. Nesse caso, além de uma potencial alteração no software, será necessária uma intervenção em cada um dos dispositivos instalados e uma mudança no hardware pode ser difícil, ou mesmo impossível, de ser realizada em campo.

Soluções baseadas em hardware, tais como nos dispositivos utilizados em soluções IoT, requerem atenção redobrada com relação à sua especificação e à qualidade/tolerância de seus

componentes para maximizar sua vida útil e minimizar falhas. A parceria entre o arquiteto de soluções em IoT e o especialista em hardware é fundamental para a correta definição dos requisitos para o dispositivo, incluindo não somente os cenários nos quais a solução funciona, mas também potenciais casos extremos de uso, como sobrecarga elétrica, eletricidade estática, proteção contra água, exposição a animais, exposição externa a fatores naturais, aterramento, acesso por pessoas não autorizadas etc.

Atualização remota do firmware

Além de todos os requisitos de hardware, os dispositivos IoT trazem uma camada adicional de complexidade que se refere ao software executado no dispositivo (*firmware*, ou software embarcado), utilizado para coletar dados, comunicar-se com a nuvem, receber e executar comandos e quaisquer outras funcionalidades requeridas pelo cenário. Mesmo sendo uma solução de software embarcado, todas as preocupações arquiteturais de software com relação aos seus atributos de qualidade continuam válidas: segurança e privacidade, escalabilidade, manutenibilidade, robustez, resiliência, tolerância a falhas, desempenho, disponibilidade e recuperação de desastres.

Como veremos no capítulo *Desafios arquiteturais de uma solução IoT*, alguns desses atributos de qualidade são especialmente complexos em um dispositivo IoT por envolverem tanto software quanto hardware para atingirem seus objetivos. Exemplo: segurança e privacidade dizem respeito tanto à criptografia necessária para o trânsito seguro de informações quanto ao potencial uso de hardware específico para o armazenamento de certificados e realização de transações criptográficas, conhecido como HSM (*Hardware Secure Module*, ou, em português, módulo de segurança de hardware).

Manter o dispositivo seguro e funcional ao longo de sua vida útil vai requerer atualizações de segurança, correção de erros e melhorias no código, que devem ser executadas remotamente. Com isso, esse

procedimento minimiza os custos de manutenção, especialmente para os dispositivos instalados em locais de difícil acesso.

1.6 Dispositivos IoT

O que é um dispositivo IoT?

Seguindo o conceito de Internet das Coisas, uma “coisa” para ser considerada parte de IoT deve possuir as seguintes características:

- Estar conectada direta ou indiretamente à internet em algum momento do tempo;
- Ter capacidade de executar software localmente, o que implica que os dispositivos devem ter ao menos um pequeno microcontrolador (MCU) que os permitam executar código – também conhecido como *firmware* – localmente;
- Ser capaz de interagir com o mundo por meio de sensores e/ou atuadores.

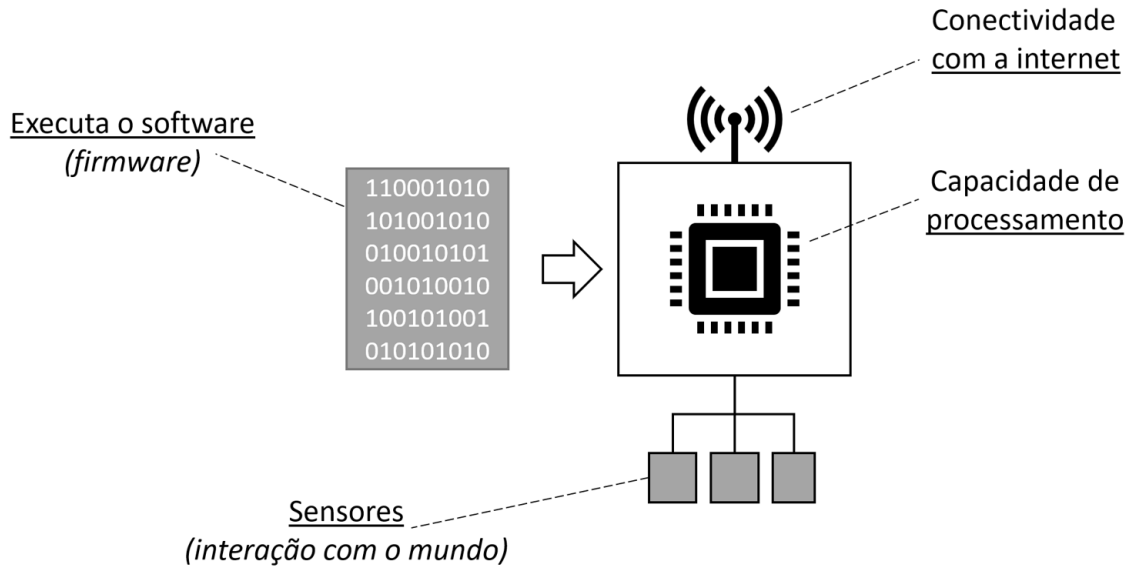

Figura 1.1: Dispositivo IoT com habilidade de executar código e conectar-se à internet.

Com essa definição abrangente, a quantidade de diferentes configurações de hardware que atendem a esses requisitos é ampla: desde pequenos MCUs alimentados por baterias até computadores com grande capacidade de processamento de imagens (GPUs) podem ser considerados dispositivos IoT. O desafio está em encontrar a melhor combinação entre o atendimento dos requisitos e o custo da solução.

Deve-se também considerar o reúso de hardware preexistente, com sensores e controladores fabricados há décadas e que precisam ser integrados à internet, tornando-se dispositivos IoT. Por exemplo, uma fabricante de laticínios que, em suas fábricas, possui enormes panelas industriais para a fermentação do leite. Tais equipamentos já utilizados por muitos anos precisarão ser modificados para permitir a coleta de telemetria com valores de seus velhos sensores de temperatura, pressão, estado etc. Nesse caso, há a necessidade de adaptadores IoT, conversores de protocolo, ou concentradores (*gateways*), que farão o meio de campo entre os dispositivos legados e a nuvem.

Teoricamente, qualquer dispositivo elétrico/eletrônico pode ser conectado, então a questão não é se o dispositivo pode ser conectado ou não, mas sim qual o valor em se conectar o dispositivo.

Em um cenário mais simples, pode-se adicionar um controlador de tomada, de forma a se operar remotamente qualquer dispositivo elétrico. Ligar ou desligar remotamente um dispositivo elétrico pode, à primeira vista, parecer um cenário de pouco valor, mas é importante ter em mente que o valor deve ser considerado do ponto de vista do usuário e não dos profissionais de TI. Assim, uma simples tomada pode adicionar valor a uma fechadura elétrica que pode ser acionada remotamente, permitindo a entrada de visitas, funcionários, entregadores etc.

1.7 Modelo de valor para IoT

Como vimos na seção anterior, embora praticamente qualquer coisa possa ser conectada e se tornar um dispositivo IoT, é necessária uma cuidadosa avaliação dos problemas que essa conexão vai resolver. Um modelo de avaliação desse valor, utilizado pela Microsoft, baseia-se em três pilares:

- Coisas (*things*) geram dados – refere-se à telemetria enviada do dispositivo para a nuvem, ou D2C (*Device to cloud*, em português, do dispositivo para nuvem);
- Epifania (ideias ou *insights*) – uma súbita compreensão de um problema ou de uma situação baseada nos dados que foram gerados pelo dispositivo;
- Ações (*actions*) – são tomadas com base na compreensão dos problemas.

Figura 1.2: Pilares do modelo de valor de uma solução IoT.

1.8 Componentes de uma solução IoT

Cada solução IoT é única e sua complexidade varia de acordo com os requisitos da solução: podem ser simples sensores remotos de baixa criticidade em que perdas de algumas mensagens não representam um problema; ou podem ser cenários onde não se admite perda de mensagens, no caso de os dados trafegados pela solução serem sigilosos ou permitirem a identificação de seus

usuários ou, ainda, se os dados não puderem ser armazenados fora do território nacional.

Como em soluções tradicionais, cenários de suporte à vida ou de máquinas que podem causar ferimentos em humanos devem ser tratados considerando padrões de qualidade condizentes com a criticidade da solução.

A figura a seguir ilustra os principais componentes de uma solução IoT, separando-os conceitualmente em camadas lógicas.

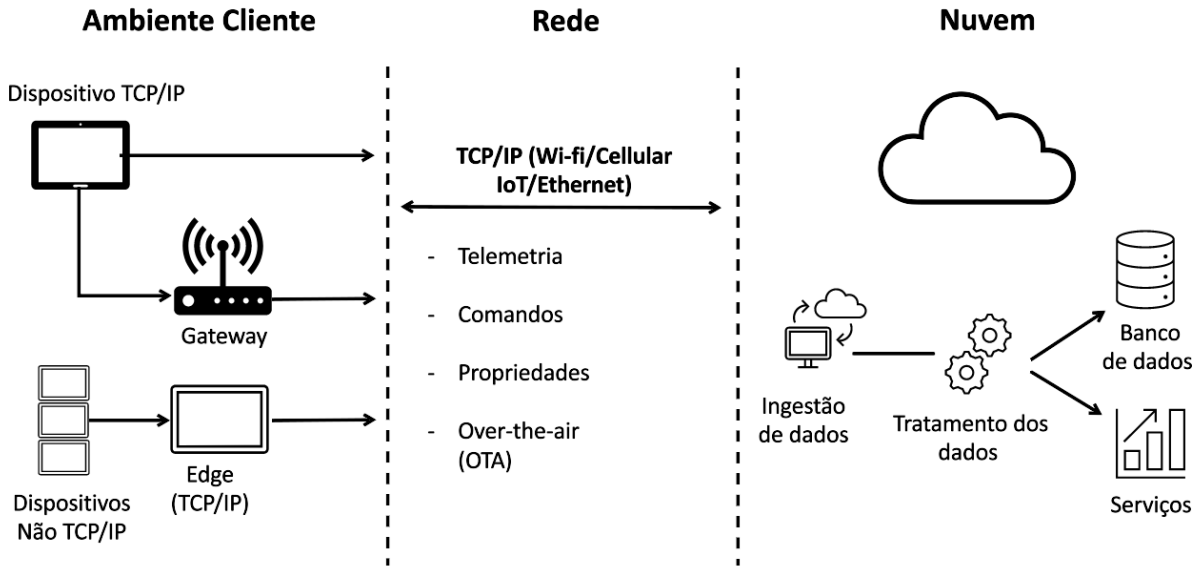

Figura 1.3: Camadas lógicas de uma solução IoT.

Ambiente Cliente / Dispositivos

Dispositivos (ou *devices*) estão presentes em soluções IoT, sendo a interface com o mundo real e usuários. Eles podem ser classificados quanto a seu modelo de trabalho, sua capacidade de processamento, consumo de energia, tamanho, utilização ou não de um sistema operacional e o tipo de rede utilizada.

Os dispositivos IoT, em sua maioria, são capazes de enviar mensagens (telemetria) para a nuvem, de reportar seu estado atual,

receber mensagens, comandos e atualizações. A combinação de dispositivos e a capacidade da nuvem possibilita a implementação de um número muito grande de cenários.

Dispositivos menores, à bateria ou de aplicações remotas, podem não ser dotados de interfaces de rede baseadas em TCP/IP, ou seja, não são capazes de se conectar à internet diretamente. Esses dispositivos geralmente se utilizam de protocolos especiais ou legados, de forma que a conversão de protocolo é necessária para que sua comunicação com a nuvem seja estabelecida. Esses conversores de protocolo são chamados de *gateways* e um único *gateway* pode suportar múltiplos dispositivos.

Um tipo especial de dispositivo IoT conhecido como *edge* (borda) possui capacidades híbridas de nuvem e *gateway*. Geralmente esses dispositivos possuem maior capacidade de processamento, alguns possuem aceleradoras gráficas (GPU) para processamento de imagens e vídeo local, assim como capacidade de execução de modelos de inteligência artificial.

Rede

Existem múltiplas formas de conectar dispositivos IoT à internet, podendo variar em diferentes combinações entre padrões de comunicação, tecnologias de comunicação e protocolos de comunicação.

A escolha do modelo de rede utilizado pela solução deve ser baseada nos resultados esperados de disponibilidade, custo, velocidade de comunicação, criticidade e consumo de energia.

Nuvem

O advento da computação em nuvem permitiu o crescimento da área de IoT por fornecer uma escalabilidade virtualmente infinita, estabilidade e grande variedade de recursos e ferramentas.

Quando os dispositivos IoT produzem dados (telemetria) e os transmitem pela internet para a nuvem, uma nova jornada se inicia, começando pela ingestão dos dados, passando por uma fase de tratamento e normalização dos dados que, então, permitirá o armazenamento destes dados em um banco de dados ou o consumo desses dados por serviços de mineração, *machine learning* ou outros sistemas distribuídos importantes para a solução.

Conclusão

O uso de IoT está cada dia mais presente em nossa rotina, seja nos meios de transporte, no ambiente de trabalho, em lojas e centros comerciais, ou dentro de nossas casas. O advento da nuvem permitiu a popularização dessas soluções devido a seu poder computacional e riqueza de recursos.

Soluções IoT não são simples e exigem conhecimento de diferentes áreas. Múltiplos profissionais farão parte do planejamento e implantação da solução. No próximo capítulo, são apresentados os profissionais envolvidos em uma solução IoT, suas responsabilidades e quais suas principais atividades em um projeto de IoT.

Capítulo 2
Profissionais em projetos de IoT

Como vimos no capítulo anterior, a criação de uma solução IoT envolve componentes que requerem conhecimentos de várias disciplinas como engenharia elétrica/eletrônica, redes de computadores, desenvolvimento de software embarcado, desenvolvimento de software para nuvem – e a lista pode ficar ainda mais longa se considerarmos soluções que demandem aprendizado de máquina e inteligência artificial.

Neste capítulo, são abordadas as principais qualificações profissionais envolvidas na criação de uma solução IoT, lembrando que a lista não é exaustiva e, conforme tecnologias evoluem, novos cargos e/ou papéis podem ser necessários, assim como os existentes podem se tornar obsoletos.

Vale ainda ressaltar que as profissões apresentadas não possuem uma relação de um para um (um profissional = um cargo), dado que um mesmo profissional pode vir a assumir diferentes responsabilidades no decorrer de um projeto, especialmente em empresas menores.

2.1 IoT e a interseção de tecnologias

Duas das principais disciplinas envolvidas na criação de soluções IoT são a Engenharia Elétrica e a Engenharia de Software. A primeira disciplina trata das definições do hardware e a segunda, de todo o software e a infraestrutura da nuvem. Qual das duas é

responsável pela interseção conhecida como IoT? A figura a seguir ilustra esse cenário.

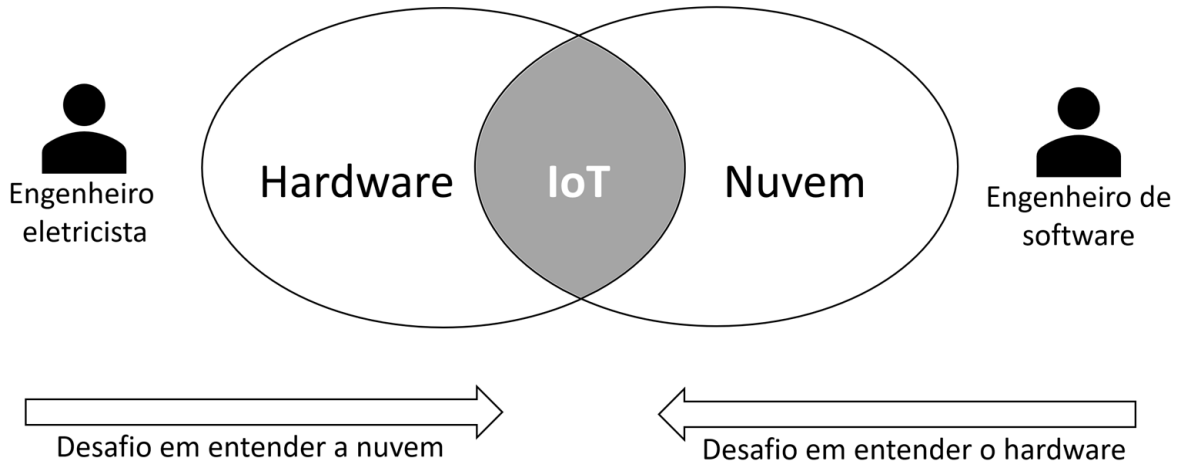

Figura 2.1: Interseção entre a disciplina de Engenharia Elétrica e a Engenharia de Software.

De forma geral, mesmo engenheiros eletricistas que estão familiarizados com o desenvolvimento de dispositivos de hardware com software embarcados têm dificuldades em entender como conectá-los à nuvem para que esses dispositivos se tornem parte da IoT.

Os engenheiros de software que atuam com soluções na nuvem, por sua vez, raramente têm a oportunidade de atuar em projetos que envolvam a utilização de hardware feito sob medida para a solução, especialmente quando esse hardware é baseado em pequenos microcontroladores.

Um grande desafio para engenheiros eletricistas e também para engenheiros de software é conhecer mais sobre as tecnologias que formam a interseção entre as duas disciplinas, para que possam melhor entender a solução de fim a fim, seus desafios e potenciais soluções. Em empresas menores, estes profissionais podem ter que atuar fora da sua áreas de especialidade para viabilizar provas de conceito e, em algumas vezes, até mesmo projetos completos.

2.2 Engenheiro(a) de hardware

A pessoa engenheira de hardware é responsável pela definição detalhada de como os dispositivos IoT se comportarão em campo. A relação entre a engenharia de hardware e a arquitetura de IoT é fundamental para que os requisitos da solução sejam suportados pelo dispositivo.

As preocupações da área de engenharia de hardware variam de acordo com a fase do projeto: prova de conceito, protótipo funcional, testes, aprovação do projeto e envio para fabricação. De forma geral, a lista a seguir engloba as principais tarefas do engenheiro de hardware no projeto:

- Discutir com o(a) arquiteto(a) de IoT os requisitos de hardware necessários;
- Identificar as condições nas quais o dispositivo estará exposto uma vez produzido: variações de temperatura, umidade, exposição à água, vento, radiação, poeira, vibração, impacto etc.;
- Identificar requisitos de segurança que possam exigir hardware específico para manipulação de certificados digitais e segredos, que, neste caso, pode requerer tipos especiais de microcontroladores ou módulos TPM (do inglês *Trust Platform Module*);
- Definir quais e que tipos de sensores e atuadores serão necessários;
- Definir como os usuários lidarão com o dispositivo: através de botões, chaves, teclas, telas, sons e quaisquer outros meios de interação necessários;
- Definir tecnologias e padrões de comunicação a serem utilizados pelo dispositivo (exemplo: wi-fi, Bluetooth, LoRa, Zigbee);
- Definir que microcontrolador ou microprocessador utilizar;
- Identificar que capacidade de memória o dispositivo necessitará para cada tipo de memória (RAM, ROM, Flash etc.);

- Construir e testar o protótipo funcional do dispositivo, em um processo iterativo em busca do seu aperfeiçoamento;
- Identificar o consumo de energia do dispositivo e buscar alternativas eficientes de gestão de energia;
- Identificar a necessidade de certificações de segurança para o dispositivo, como certificação para uso automotivo (MISRA), padrões de segurança IEC (IEC 61508 SIL 4 e IEC 62304 Class C), fabricação sem utilização de chumbo, materiais que podem entrar em contato com alimentos etc.;
- Construir a lista de componentes (do inglês BOM, *Bill Of Materials*) que serão utilizados na produção do dispositivo, com o intuito de minimizar o seu custo de produção;
- Definir critérios de testes e aceite para os dispositivos que serão fabricados;
- Criação do projeto final do dispositivo e instruções para fabricação, incluindo o diagrama esquemático e placas de circuito impresso.

Dessa forma, a engenharia de hardware é responsável por todo o processo desde a criação do protótipo até o produto final. Em diversas situações onde não existe o engenheiro de hardware, empresas e consultorias de engenharia terceiras podem ser contratadas para a realização dessas etapas. A proliferação de kits de desenvolvimento com diversos tipos de microcontroladores facilita a criação de protótipos por pessoas com algum conhecimento em eletrônica.

Notas

- International Electrotechnical Commission (IEC) é uma organização internacional responsável pela elaboração e publicação de padrões para tecnologias elétricas ou eletrônicas. Para mais informação, acesse: https://iec.ch/.
- Para conferir exemplos dos kits de desenvolvimento citados, acesse https://store.arduino.cc/usa/new-home/education.

2.3 Engenheiro(a) de software embarcado

O(a) engenheiro(a) de software embarcado é responsável pelo código que será executado no dispositivo, geralmente escrito em uma linguagem de programação de baixo nível, normalmente em C, C++ ou Assembler. Recentemente, alternativas baseadas em Python ou mesmo Rust tornaram-se comuns. Assim como no caso da engenharia de hardware, sua parceria com a arquitetura de IoT é extremamente importante.

O conhecimento da pessoa engenheira de software embarcado vai muito além da linguagem de programação. Esse profissional precisa conhecer o hardware utilizado, a arquitetura do processador e seu conjunto principal de instruções para, assim, otimizar o uso de recursos e maximizar o desempenho do dispositivo. Suas responsabilidades também incluem:

- Discutir com o(a) arquiteto(a) de IoT os requisitos de software da solução;
- Definir se o código da solução fará uso de um sistema operacional de tempo real (do inglês RTOS, *Real Time Operating System*);
- Identificar dependências de bibliotecas de terceiros (bibliotecas de rede, dispositivos, criptografia, JSON parsing etc.), além de componentes de software adicionais a serem criados;
- Desenvolver o código a ser utilizado pelo protótipo funcional do dispositivo;
- Desenvolver testes de unidade para todo código produzido;
- Testar o código no protótipo funcional do dispositivo. O uso de simuladores pode agilizar o processo de desenvolvimento, mas o acesso ao hardware do dispositivo é crucial para que os testes sejam próximos à realidade;
- Verificar se o código produzido atende aos requisitos de segurança da solução;
- Verificar se o código produzido atende aos requisitos funcionais e não funcionais da solução, assim como a normas exigidas

que podem ser comprovadas por meio de certificações;
- Gerar a versão final do código embarcado (conhecido como *firmware*) para testes e produção.

Espera-se que a pessoa da área de engenharia de software embarcado tenha a capacidade de extrair o máximo do hardware utilizado para que os objetivos da solução sejam alcançados.

Devido a todas as características únicas do engenheiro de software embarcado, esse profissional é bastante raro no mercado de trabalho, o que o torna valorizado pelas empresas. A contratação de consultorias tem se tornado a solução para muitas empresas que estão em busca desse perfil profissional no mercado, mas têm dificuldade em encontrá-los.

2.4 Engenheiro(a) de segurança

Embora não seja comum a separação entre os cargos da engenharia de software e da engenharia de segurança, é crucial que a segurança da aplicação seja considerada como aspecto fundamental em seu design, tanto para hardware quanto para software. Assim, a pessoa que trabalha na área de engenharia da segurança é responsável por garantir que os requisitos de segurança sejam atendidos pela solução. Suas responsabilidades também incluem:

- Discutir com o(a) arquiteto(a) de IoT os requisitos de segurança da solução;
- Mapear a superfície de ataque de cada elemento da solução, seja esse um dispositivo, *gateway*, conexão de internet ou recurso na nuvem;
- Definir o nível de segurança de acesso físico e lógico dos dispositivos;

- Identificar a solução mais segura para armazenamento de dados no dispositivo, podendo ser em arquivos locais do sistema ou na memória do microcontrolador;
- Identificar a solução mais segura para armazenamento local de credenciais nos dispositivos, sejam esses certificados digitais ou chaves primárias e secundárias. Entre as possibilidades de armazenamento, podemos ter arquivos locais em disco, memória do microcontrolador ou dispositivos externos de segurança;
- Definir como os dispositivos protegerão o canal de comunicação com a nuvem;
- Definir critérios de segurança e aceitação para os dispositivos a serem fabricados;
- Definir táticas de proteção, detecção, reação e recuperação dos dados em caso de roubo de dispositivos ou tentativa de ataque à solução.
    - Proteção: adotar meios de proteção que restrinjam o acesso à informação contida em elementos da solução.
    - Detecção: a partir de anomalias em um ou mais elementos da solução, pode-se identificar a possível atuação de um usuário mal-intencionado.
    - Reação: pode variar entre reportar a tentativa e continuar a operação, até mesmo invalidar credenciais e bloquear o acesso do dispositivo ao restante da solução.
    - Recuperação: recuperar a integridade dos elementos da solução para retornar a sua operação.

Especialmente em cenários de soluções críticas ou produtos que serão vendidos a uma grande quantidade de clientes, essas atividades são cruciais para minimizar danos aos dispositivos, acesso indevido aos dados e – ultimamente – danos à imagem da empresa por colocar no mercado produtos pouco seguros. Assim sendo, mesmo que não exista uma pessoa engenheira de segurança no time de desenvolvimento, essas tarefas devem ser executadas.

2.5 Engenheiro(a) de software de nuvem

O(a) engenheiro(a) de software de nuvem é responsável pelo código, configuração e serviços que serão executados na nuvem. Embora estejamos genericamente chamando este cargo de engenheiro de software de nuvem, suas atividades podem envolver diversos profissionais de diferentes áreas, desde garantir a conexão do dispositivo com o serviço de nuvem, armazenar e tratar os dados enviados, até mesmo orquestrar comandos que serão processados pelos dispositivos e gerenciar o estado do dispositivo (por exemplo, versão do *firmware*, estado desejado de propriedades, certificados digitais). Suas responsabilidades também incluem:

- Discutir com o(a) arquiteto(a) de IoT os requisitos da solução;
- Criar o serviço de gerenciamento e conexão para comunicação de dispositivos IoT;
- Definir rotas de processamento para os dados, tais como: rotas para serviços de streaming para análise dos dados, rotas para armazenamento dos dados e rotas para o disparo de funções customizadas;
- Criação e codificação de funções e rotinas para o tratamento dos dados;
- Criação e configuração de serviço para o armazenamento dos dados;
- Criação e configuração de serviço para análise dos dados via streaming;
- Criação de integrações híbridas entre um serviço na nuvem e quaisquer sistemas externos, tais como CRM (*Customer Relationship Management*) e ERP (*Enterprise Resource Planning*);
- Criação de testes a fim de validar a integração entre sistemas, atendimento de requisitos funcionais e não funcionais (atributos de qualidade), e testes de carga.

Essa lista de responsabilidades pode variar de acordo com as características do projeto, do provedor de nuvem utilizado, do

desenho da solução (infraestrutura – plataforma ou software como serviço) e da criticidade da solução.

2.6 Arquiteto(a) de IoT

De acordo com o SEI, Software Engineering Institute (https://www.sei.cmu.edu/our-work/software-architecture/), cabe à pessoa arquiteta de software criar uma solução efetiva de software que suporte o processo de negócio, atendendo a seus atributos de qualidade (como segurança, escalabilidade, disponibilidade, manutenibilidade, desempenho, usabilidade etc.) e que permita sua evolução no futuro.

Os atributos de qualidade devem ser considerados através de toda a solução, dado que seu desempenho total será igual a seu elo mais fraco. Exemplo: a segurança total da solução vai desde o dispositivo IoT, sua conexão com a internet, aos dispositivos de rede com os quais se comunica, às credenciais utilizadas, à proteção dos serviços de nuvem e a outros componentes da solução. Dessa forma, de pouco adiantaria ter a rede e o serviço de nuvem bem protegidos contra ataques se o dispositivo IoT puder ser comprometido e se suas credenciais estiverem armazenadas de forma aberta (não criptografada).

Nota

Um caso curioso com um termostato em um cassino nos Estados Unidos, também publicado na HackRead (WAQAS, 2018), mostra como dispositivos IoT podem abrir brechas de segurança para toda uma rede.

Neste cassino, um termostato instalado em um aquário na recepção estava conectado à rede wi-fi para auxiliar no controle da temperatura da água.

No entanto, a conexão desse dispositivo com a rede do cassino era pouco segura, o que permitiu que invasores capturassem a sessão IP e invadissem a rede. Para piorar, as redes internas do cassino não eram segregadas da rede wi-fi à qual este dispositivo estava conectado.

Como resultado, os invasores tiveram acesso à base de dados de apostadores de altas quantias, que foi então roubada. Embora não tenham sido publicados maiores detalhes sobre a natureza do ataque ou do impacto do roubo, esse exemplo deixa claro que invasores sempre buscarão o ponto mais frágil na infraestrutura, que, nesse caso, era um termostato IoT pouco seguro e que serviu de porta de entrada.

A identificação e gestão dos *tradeoffs* é também de responsabilidade do arquiteto de IoT: o profissional precisa compreender que a tomada de decisão técnica pode afetar dois ou mais atributos de qualidade e precisa estar preparado para avaliar seu impacto na solução. Exemplo: incrementar a segurança pode impactar o desempenho da solução por adicionar carga adicional ao processamento, mas pode ser um *tradeoff* válido, dada a necessidade de proteger os dados.

Há diversos métodos para guiar o(a) arquiteto(a) de software em suas decisões em um projeto de IoT, entre eles (CLEMENTS *et al.*,

2010):

- ATAM (do inglês *Architecture Tradeoff Analysis Method*), para identificação e gestão de *tradeoffs*;
- ADD (do inglês *Attribute-Driven Design*), para design da arquitetura de forma iterativa por meio do uso de diretrizes (*drivers*) como entradas para a produção da arquitetura;
- Documentação através de Visões (do inglês *View*), onde cada visão é uma representação de um conjunto de elementos do sistema e relacionamentos entre eles.

As responsabilidades do arquiteto de IoT também incluem:

- Capturar e documentar os requisitos da aplicação (funcionais e não funcionais);
- Discutir com outros profissionais os requisitos de cada área da solução (segurança, nuvem, hardware etc.);
- Identificar e documentar os diversos componentes da solução (desde os dispositivos até os serviços da nuvem e sua integração com possíveis externos), dados os requisitos da aplicação;
- Criar e manter o plano de comunicação do projeto, mantendo todos os envolvidos cientes do andamento, riscos e decisões;
- Identificar e documentar decisões de *tradeoffs*;
- Garantir que os requisitos e atributos de qualidade sejam atendidos;
- Acompanhar o andamento do projeto para garantir a aderência com o design original da solução;
- Planejar o custo operacional da solução: fabricação e distribuição dos dispositivos IoT, custo de consumo dos dispositivos IoT, consumo de recursos na nuvem (ingestão de dados, armazenamento de dados históricos, provisionamento de recursos etc.);
- Planejar a escalabilidade e elasticidade de recursos na nuvem para garantir sua funcionalidade durante períodos de maior consumo da aplicação, assim como minimizar custos em períodos de menor uso.

- Planejar alta disponibilidade da solução, incluindo distribuição geográfica de recursos, redundância e recuperação após desastres;
- Classificar os tipos de dados persistidos pelo seu uso, de acordo com as leis de soberania de dados – por exemplo, a Lei Geral de Proteção de Dados, originalmente inspirada na regulamentação europeia de proteção a dados chamada GDPR (*General Data Protection Regulation*);
- Desenvolver planos de testes que validem a integração entre os diferentes componentes da solução com base nos cenários de uso da aplicação;
- Aprovar a solução final antes da entrega ao cliente.

NOTA

Uma das mais importantes tarefas do(a) arquiteto(a) de IoT é a de identificar e documentar as decisões de *tradeoffs*.

Normalmente, é muito clara a necessidade de identificar os *tradeoffs* entre os atributos de qualidade e tomar decisões baseadas nesta análise, tais como: segurança *versus* tempo de resposta e alta disponibilidade versus custo da solução.

Uma tarefa que é tão importante quanto a análise de *tradeoffs*, mas que é normalmente esquecida, é a documentação dessas decisões. Em tempo de projeto, enquanto o time está trabalhando para criar uma solução IoT, lembrar dos porquês por trás de uma decisão é uma tarefa relativamente simples. O problema aparece quando, no futuro, as decisões tomadas no projeto são questionadas: o time de projeto já não está mais junto, pessoas tendem a esquecer os detalhes e profissionais podem ter até mesmo deixado a empresa.

Por este motivo, documentar a razão por trás de cada decisão de *tradeoff* é fundamental: deve-se documentar as alternativas avaliadas, as razões que levaram o(a) arquiteto(a) a decidir por uma delas e as razões para que as demais alternativas tenham sido postas de lado.

2.7 Outras áreas envolvidas

Neste capítulo, abordamos os profissionais mais relevantes no desenvolvimento de uma solução IoT, mas, como mencionamos anteriormente, em empresas menores, os profissionais podem ter

vários papéis simultaneamente. Da mesma forma, empresas maiores podem ter profissionais dedicados a cada função, podendo também optar pela terceirização.

Algumas áreas não mencionadas em detalhes incluem:

Gerenciamento de projetos: responsável pelo andamento do projeto, comunicação, riscos, cronograma, gestão das pessoas, em nível não técnico.

Gerenciamento de produto: responsável pela estratégia do produto, identificação das funcionalidades necessárias, pesquisa de mercado, sendo ultimamente responsável pelo sucesso do produto.

Engenharia de redes: responsável pela definição, implementação e validação da estrutura de comunicação entre os dispositivos e a nuvem, e entre os componentes da nuvem.

Conclusão

Como visto neste capítulo, por conta de sua complexidade e riqueza em detalhes, soluções IoT exigem a participação de muitas pessoas exercendo diferentes responsabilidades com o intuito de entregar um produto de qualidade, robusto, seguro e funcional.

Para muitos profissionais, o maior desafio em começar a trabalhar com IoT está em entender o funcionamento de dispositivos IoT, pois esta área é aquela que mais se diferencia das demais soluções de tecnologia com as quais têm experiência.

Desenvolver soluções baseadas na comunicação entre uma pessoa com um serviço (por exemplo, página web) ou entre um serviço com outro serviço (por exemplo, serviço distribuído) são modelos com que muitos profissionais já estão habituados a lidar no dia a dia, mas desenvolver soluções centradas na comunicação entre múltiplos dispositivos e um sistema distribuído exige a adoção de novas práticas, assim como entender os limites e a capacidade destes dispositivos.

No próximo capítulo, serão detalhados os dispositivos IoT, seus possíveis usos, tipos, modelos de desenvolvimento e integração com sensores e atuadores.

Capítulo 3
Dispositivos IoT

Parte fundamental de uma solução IoT, os dispositivos (ou as “coisas” mencionadas no termo Internet das Coisas) são as unidades acessíveis a nós fisicamente em soluções IoT, ou seja, dispositivos físicos que interagem com o mundo real. Esses dispositivos variam quanto a seu tamanho e capacidade de processamento: desde pequenos sensores alimentados por baterias até dispositivos robustos, com capacidade de processamento suficiente para reconhecimento facial e de objetos.

O uso de dispositivos IoT é tão amplo que entender toda a sua extensão e aplicabilidade pode ser uma tarefa intimidadora. A seguir, são apresentados exemplos de dispositivos e possíveis cenários de uso:

1. **Dispositivo IoT com sensores de temperatura e umidade**: comumente utilizados em diversos cenários de automação e coleta de telemetria para análise do tempo, tais como: centrais de ar-condicionado, para automação predial; instalações em florestas, para identificação de mudanças do tempo; fazendas, para controle de grãos estocados; ou em fornos industriais, para controle de seu funcionamento.

2. **Dispositivo IoT com sensor de vibração**: utilizados na monitoração do nível de vibração de um equipamento para detectar a necessidade de manutenção preventiva, minimizando paradas não programadas. Esse modelo de dispositivo é utilizado em elevadores, escadas rolantes, tratores, motores de embarcações e máquinas industriais.

3. **Dispositivo IoT com microfone**: próprios para identificação de variação no volume e no padrão sonoro gerados por uma máquina em funcionamento, com o objetivo de detectar

anomalias que exijam manutenção. Ao contrário de sensores mais simples, esse tipo de identificação exige um maior poder de processamento para execução de um modelo de inteligência artificial que representa o padrão normal de uso da máquina. Outro uso cada vez mais popular de dispositivos IoT com microfone são os assistentes por voz (por exemplo, Amazon Alexa, Google Home, Apple Siri, Microsoft Cortana), que possuem um modelo de inteligência artificial bastante simplificado, focado em identificar apenas suas respectivas palavras-chaves (mais conhecidas como *wake up words*) e direcionar todo o conteúdo do comando de voz para a nuvem.

4. **Dispositivo IoT com sensor de umidade de solo**: específicos para uso tanto em estufas de plantas quanto em monitoração do solo em fazendas, indica a condição do solo e a necessidade de irrigação. Geralmente são utilizados em conjunto com sensores de minerais importantes para as plantas que estão monitorando. Os modelos destes sensores variam desde sensores resistivos para uso doméstico até sensores capacitivos de uso industrial próprios para suportar condições climáticas extremas. Seu uso também pode ser aplicado em qualquer cenário em que seja necessário controlar a umidade de algum item sensível a mudanças, como vinhos, alimentos, medicamentos, instrumentos musicais.

5. **Dispositivo IoT com sensores de telemetria para veículos**: os veículos modernos já possuem dezenas de sensores que auxiliam o seu funcionamento. Exemplos de sensores são o de temperatura do motor, do líquido de arrefecimento, temperatura e nível do óleo, nível do combustível no tanque, pressão dos pneus, temperatura na cabine dos passageiros, funcionamento do sistema de injeção eletrônica, entre muitos outros. Adicione à lista capacidades mais recentes como detecção de acelerações e freadas bruscas, velocidade instantânea do veículo, localização geográfica, câmeras e microfones. Com todos esses recursos, os cenários de uso são muito variados:

seguradoras estão utilizando dispositivos IoT para identificar padrões de uso dos motoristas para decidir o valor da apólice de seguro; empresas de ônibus, de entregas, locadoras e táxis também se utilizam desses dispositivos para saber onde os veículos de sua frota estão e como estão sendo utilizados. Existem até mesmo cenários em que pais utilizam esta solução para monitorar a maneira como seus filhos mais jovens estão dirigindo.

6. **Dispositivo IoT com monitor cardíaco**: estes dispositivos utilizam-se de sensores que são colocados sobre a pele do usuário e são capazes de detectar o batimento cardíaco e sua frequência. Eles podem ser utilizados tanto em conjunto com relógios de pulso quanto com sensores que ficam ao redor do tórax do usuário. O cenário de uso mais comum é o acompanhamento do batimento cardíaco durante exercícios físicos, mas alguns modelos são precisos e sensíveis o bastante para fins médicos, como o acompanhamento de pacientes no período pós-operatório.

A seguir, apresentamos três diferentes perspectivas de classificação de dispositivos IoT:

- **Sensores e atuadores**: como os dispositivos interagem com o mundo;
- **Classes dos dispositivos**: microcontroladores *versus* microprocessadores.
- **Sistemas embarcados**: *bare-metal*, RTOS e interpretadores.

3.1 Sensores e atuadores

Sensores e atuadores permitem que um dispositivo IoT interaja com o mundo real. O mais simples dos sensores são os botões. Encontrados em um imenso número de dispositivos, eles

transformam uma ação mecânica de pressionar uma peça em uma informação digital.

Assim como um botão, dispositivos IoT podem ter uma variedade imensa de sensores. Alguns exemplos (não exaustivos) podem ser:

- Luz – identificar presença ou ausência de luz, além de sua intensidade;
- Umidade;
- Temperatura;
- Nível de líquidos;
- Identificadores de cor;
- Gás – praticamente todos os tipos de gases possuem um sensor dedicado, sendo os de CO_2 os mais comuns;
- Sons;
- Distância;
- GPS (*Global Positioning System*);
- Movimento;
- Acelerômetros.

Cada um desses sensores (ou vários deles ao mesmo tempo) podem ser conectados a um dispositivo IoT, transformando seu valor de leitura em informações digitais que serão enviadas à nuvem.

Os atuadores, por outro lado, transformam os dados digitais em algo que podemos perceber no mundo real. O exemplo mais comum de um atuador é um simples LED que acende ou apaga para nos informar algo.

Além dos LEDs, alguns exemplos (não exaustivos) podem ser:

- Telas LCD;
- Buzinas (*buzzer*);
- *Drivers* de motores;
- Válvulas e bombas de líquidos;
- Relés (controle de cargas elétricas).

3.2 Classes dos dispositivos

Dentre os diversos fatores que diferenciam classes de dispositivos IoT, dois dos mais importantes são a capacidade de processamento e a memória disponível para operações.

A capacidade de processamento definirá a velocidade e a quantidade de operações paralelas que um dispositivo poderá realizar. Dispositivos com processadores mais simples e restritos ainda são capazes de estabelecer uma conexão segura com a internet e enviar dados de telemetria, mas aplicações em tempo real com controle simultâneo de múltiplos sensores exigirão dispositivos com maior capacidade de processamento.

Quando falamos em dispositivos IoT, a memória pode ser classificada entre:

- RAM (*Random Access Memory*): memória volátil para processamento das operações do dispositivo;
- Flash ou ROM (*Read-only Memory*): memória onde o código do dispositivo e informações não voláteis são armazenadas.

A capacidade e a velocidade de escrita dessas memórias podem variar de um dispositivo para outro.

NOTA

Outros fatores importantes que devem ser considerados na escolha do dispositivo são:

- Tipo de conectividade de rede ou entre dispositivos;
- Existência de módulos de segurança;
- Capacidade de processamento gráfico (muito importante para execução de módulos de inteligência artificial baseados no reconhecimento de padrões em imagens);
- Quantidade de portas de entrada e saída de uso geral (do inglês GPIO, – *Generic Purpose Input and Output*).

3.3 Microprocessadores versus microcontroladores

MPU (do inglês *Microprocessor Unit*), ou microprocessadores, são universalmente conhecidos como a base de computadores pessoais e servidores que normalmente executam um sistema operacional como Unix, Linux ou Windows. Microprocessadores requerem componentes externos para seu funcionamento, como memória RAM, dispositivos de armazenamento de longo prazo (como um disco rígido), além de conexão com monitores, placas de som e quaisquer outros periféricos necessários.

Na grande maioria dos casos, os microprocessadores possuem um sistema operacional que se encarrega de gerenciar todos os aspectos do hardware, abstraindo para o usuário toda essa complexidade. Dispositivos como estes podem fazer parte de soluções IoT e possuem, geralmente, alta capacidade de processamento e memória.

Single-board computer: ainda baseados em MPU, dispositivos em uma única placa, como o Raspberry Pi (www.raspberrypi.org) e Latte Panda (www.lattepanda.com), são capazes de executar sistemas operacionais como Linux e Windows, comumente conhecidos como versões de Linux ou Windows *embedded* (em português, embarcado). Na maioria dos casos, seus usuários não têm acesso a modificar os componentes do dispositivo, como aumentar a quantidade de memória ou realizar melhorias no processador. Dentre as vantagens de se utilizar pequenos computadores como estes, temos:

- GPIO – existência de portas de entrada e saída de uso geral (GPIO) que podem ser diretamente conectadas a sensores e atuadores.
- Baixo custo e consumo de energia – quando comparados a computadores tradicionais e notebooks;
- Tamanho compacto – possibilidade de que estes dispositivos sejam integrados ao equipamento que eles controlarão devido ao seu pequeno tamanho.

A figura a seguir apresenta os níveis de camadas de uma solução rodando em um microprocessador. A camada mais alta corresponde à aplicação sendo executada em um sistema operacional, que faz uso do hardware que provê recursos como GPIO, conexão *ethernet*, *Bluetooth* e wi-fi.

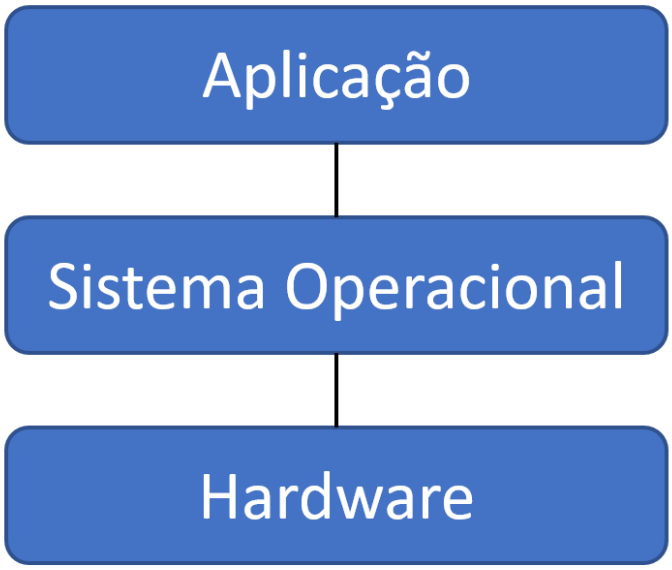

Figura 3.1: Camadas lógicas em uma solução embarcada em um microprocessador.

MCU (do inglês *Microcontroller Unit*), ou microcontrolador, ao contrário dos microprocessadores, possuem memória, capacidade de processamento, temporizadores e interfaces, todos dentro de um mesmo componente. É normal encontrar MCUs com velocidade de processamento iguais ou inferiores a 100 MHz e memória RAM de 256 KB. Mas a configuração de hardware pode variar por existirem centenas de configurações de MCU disponíveis no mercado.

Microcontroladores não utilizam sistemas operacionais convencionais, tais como Windows e Linux. Para esses dispositivos, são adotadas outras alternativas como uma versão mais compacta de sistema operacional, como RTOS (*Real Time Operating System*) ou apenas o código da aplicação (*bare-metal*) – RTOS e *bare-metal* são detalhados ainda neste capítulo.

A figura a seguir, extraída do manual do microcontrolador ATMega328 (utilizado no Arduino Uno – https://www.arduino.cc/en/Main/arduinoBoardUno), apresenta o diagrama em blocos de todos os componentes existentes internamente. Note que todos os componentes essenciais já estão inclusos no microcontrolador, como o módulo de gerenciamento de

energia, temporizador (timer), memória (SRAM, EEPROM, *Flash*), CPU, circuitos de comunicação com diversos protocolos (USART, SPI, TWI) e o GPIO (*Port* D, B e C).

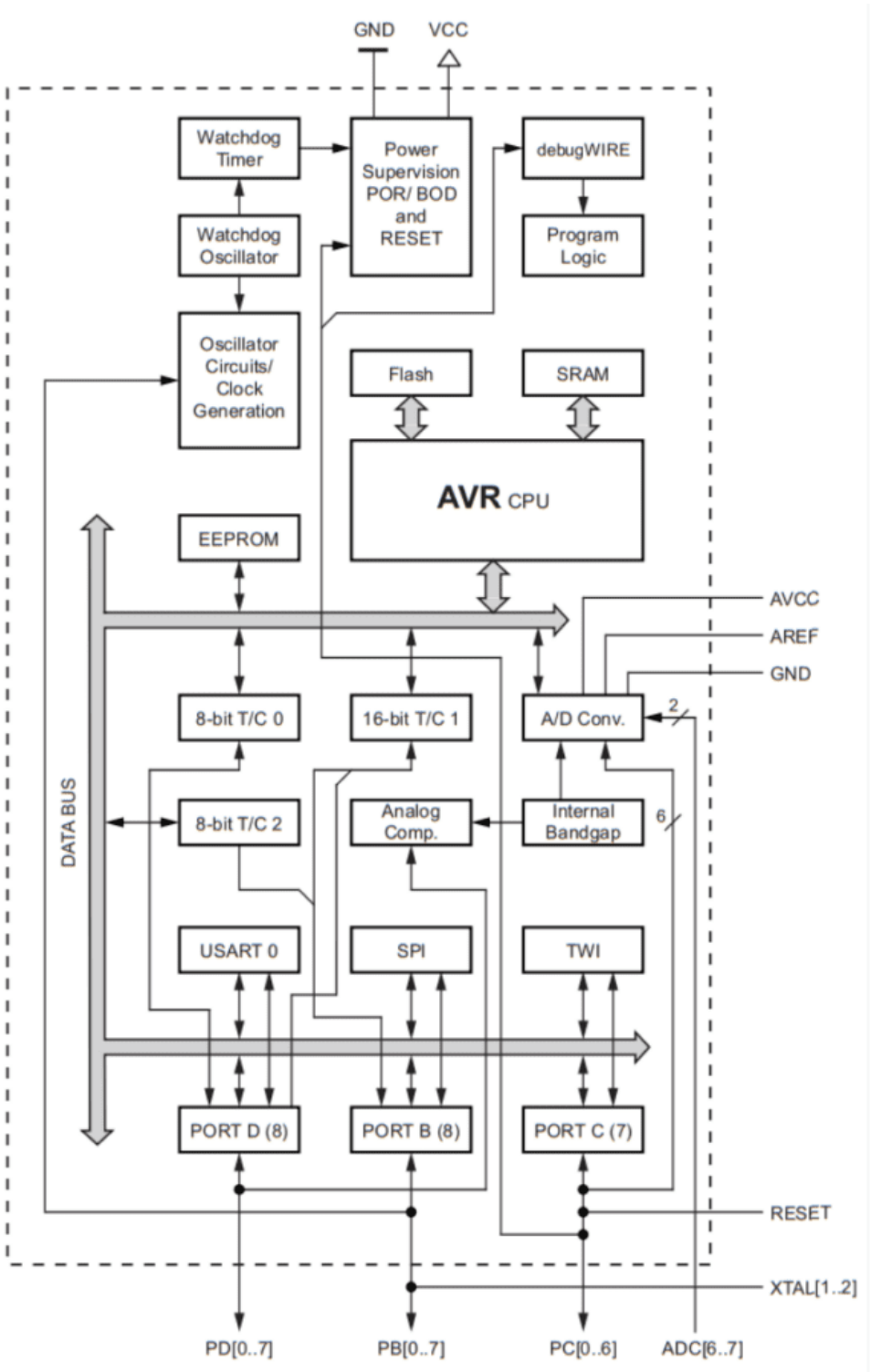

Figura 3.2: Diagrama em blocos do ATMega328 (ATMEL CORPORATION, 2015).

Os benefícios em utilizar microcontroladores são:

- Baixo custo – os MCUs estão disponíveis em centenas de configurações, permitindo que a pessoa engenheira de hardware selecione o MCU que possua exatamente a capacidade de processamento e memória para a solução;
- Menor consumo de energia – sua baixa velocidade de processamento e capacidade de memória (em comparação com MPU) resulta em baixo consumo de energia, o que possibilita vários cenários em que os dispositivos são alimentados por baterias ou por energia solar. Além disso, quase todos os MCUs possuem modos de economia de energia em que partes do seu circuito podem ser ligadas e desligadas conforme a necessidade;
- Maior confiabilidade – dispositivos baseados em MCUs podem ser utilizados em ambientes hostis com grandes variações de temperatura, vibração, ruído eletromagnético, poeira etc. Em cenários como esses, um MPU com componentes externos possui um número maior de pontos de falha, enquanto um MCU, por ter a maioria dos periféricos internamente, sofre e falha menos.

3.4 Sistemas embarcados para IoT (Embedded Development)

Geralmente, os dispositivos IoT baseados em MCUs não possuem um sistema operacional convencional e cabe ao sistema embarcado controlar cada aspecto do hardware: interrupções, processador, alocação e liberação de memória, entradas e saídas de uso geral (GPIO), temporizadores, velocidade do clock do processador etc.

As opções para um dispositivo IoT baseado em MCU resumem-se a:

- *Bare-metal* – o código da sua aplicação controlará todos os aspectos do dispositivo. A maioria das implementações nessa categoria utiliza a linguagem de programação C e/ou C++, dado o seu desempenho, versatilidade, portabilidade e disponibilidade de bibliotecas para diversos MCUs e periféricos;
- RTOS – adoção de um sistema operacional em tempo real, que normalmente utiliza C e/ou C++;
- Interpretadores – o uso de interpretadores, presentes na memória do dispositivo e que gerenciam o código de alto nível, que é então executado.

Bare-metal

Aplicações desenvolvidas em *bare-metal* controlam cada aspecto do hardware dando aos desenvolvedores controle total sobre ele: controle de cada ciclo de *clock* do processador e de cada bit da memória.

O uso de *bare-metal* é comum em dois cenários:

1. Cenários críticos com limitações extremas de hardware, como no desenvolvimento de soluções que tenham acesso restrito a energia (por exemplo, aparelho de marca-passo, estações meteorológicas movidas a energia solar).
2. Produção em alto volume de dispositivos, cujo custo por unidade impacta o resultado e o preço do produto no mercado (por exemplo, brinquedos, eletrônicos descartáveis, relógios).

A figura a seguir ilustra o código da aplicação, também conhecido como sistema embarcado, sendo executado diretamente no hardware do microcontrolador – o que permite controle total dos recursos.

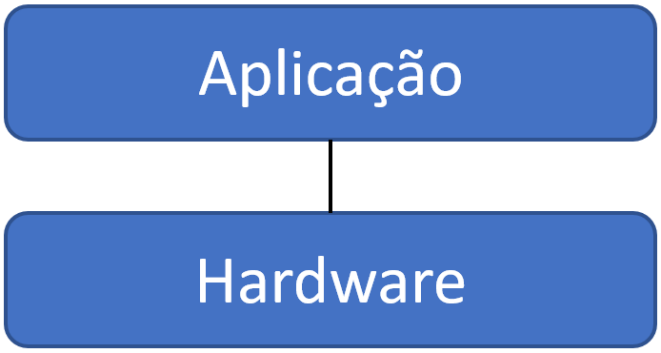

Figura 3.3: Camadas lógicas de um sistema embarcado em um microcontrolador.

Se, por um lado, os desenvolvedores têm mais controle sobre o hardware e conseguem criar soluções enxutas usando *bare-metal*, por outro, a complexidade de se escrever, testar e manter essas aplicações é maior.

Técnicas como *polling* (onde o código é executado em um loop infinito) ou máquinas de estado são normalmente utilizadas em conjunto com interrupções de hardware para o controle de aplicações em *bare-metal*.

RTOS

Real Time Operating Systems (RTOS, ou, em português, Sistemas Operacionais de Tempo Real) são a alternativa para dispositivos com microcontroladores, pois, ainda que sejam pequenos em tamanho e na sua utilização de recursos, sua lista de recursos é vasta: gerenciam a execução das *threads* (criação, suspensão, execução, finalização e preempção), assim como facilitam o uso de *multi-threading*; gerenciam o consumo de memória (alocação e liberação); controlam semáforos, *mutexes*, comunicação entre processos e gerenciamento de filas de processamento e de mensagens.

Ao contrário dos sistemas operacionais convencionais, os RTOS não são sistemas operacionais que exigem um processo de

instalação: eles funcionam como uma dependência da aplicação. Durante o processo de build, os componentes do RTOS passam a fazer parte da aplicação em um modelo de composição.

A figura a seguir ilustra uma solução onde a aplicação possui dependência para um RTOS, que gerencia o acesso e uso do hardware em um dispositivo IoT.

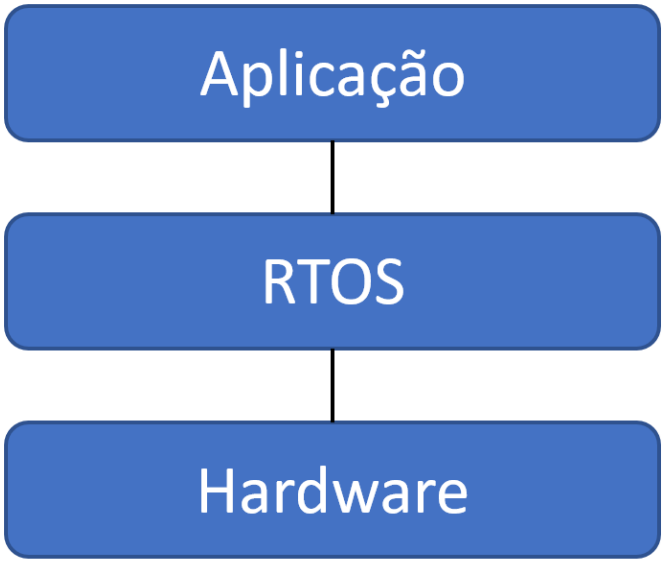

Figura 3.4: Camadas lógicas de um sistema embarcado em um microcontrolador utilizando RTOS.

Com todos estes recursos à disposição, o desenvolvedor pode focar a sua atenção no seu código, nas *threads* em que este será executado e determinar a prioridade de cada *thread*. O RTOS se encarrega de gerenciar os recursos do dispositivo e sua correta execução.

Uma das principais características de um RTOS é o determinismo, que tem relação ao nome “tempo real”: em um sistema *bare-metal*, a execução em tempo real depende da habilidade do desenvolvedor de criar uma combinação entre loops, máquinas de estado e interrupções de hardware para garantir que eventos críticos sejam atendidos dentro do tempo esperado. Como pode-se imaginar, isso não é uma tarefa simples. Com o uso de um RTOS, tarefas críticas

podem ser criadas com maior prioridade e, assim que necessário, responder a um evento crítico.

O RTOS pode fazer a preempção (suspender e salvar em memória o estado de quaisquer *threads* em execução) e dedicar recursos a uma tarefa mais crítica. O determinismo vem da capacidade de um RTOS de executar tais tarefas em um tempo determinado de forma consistente.

Embora exista uma curva de aprendizado para se conseguir os benefícios de um RTOS, sua vantagem é evidente em vários cenários de negócio mais complexos, como em automóveis, equipamentos industriais, equipamentos médicos, dispositivos de suporte à vida, entre outros.

Interpretadores

Um dos modelos mais recentes de desenvolvimento para dispositivos baseados em microcontroladores é o uso de interpretadores – meio termo entre o desenvolvimento *bare-metal* e o uso de um RTOS. Uma de suas vantagens é a possibilidade de uso de linguagens de programação como Python ou C#.

Existem modelos de hardware que já são comercializados com interpretadores instalados por padrão, como o *Adafruit Circuit Playground* (https://www.adafruit.com/product/3333) ou o *BBC Microbit* (https://www.microbit.org/). Esses kits de desenvolvimento são voltados para estudantes e *hobbistas*.

Também existe a possibilidade de se utilizar interpretadores em modelos de hardware que não os possuem como padrão. Para Python, duas vertentes têm se consolidado neste universo: *Micro Python* (criado por uma organização sem fins lucrativos de mesmo nome) e *Circuit Python* (uma vertente criada pela Adafruit, uma indústria dos Estados Unidos que é uma das maiores contribuidoras do projeto original). E para C#, a Microsoft suporta o desenvolvimento do *.NET nanoFramework*.

A figura a seguir ilustra uma solução onde a aplicação utiliza um interpretador, que traduz o código escrito em uma linguagem de alto nível para um modelo executável no hardware do dispositivo IoT.

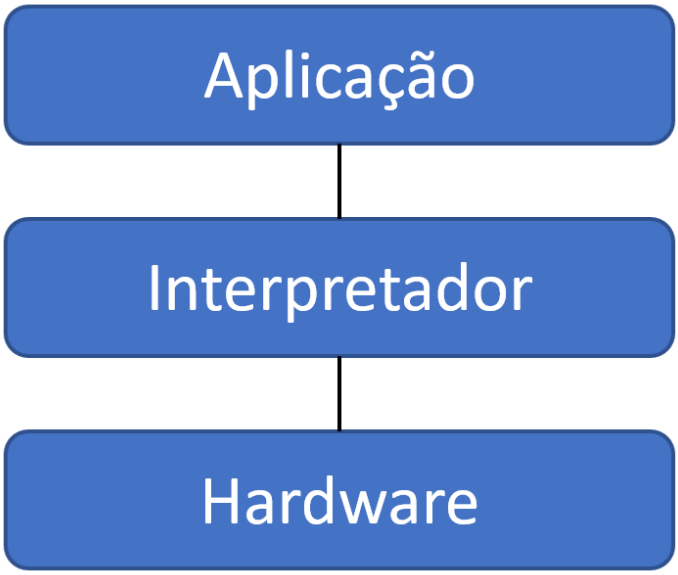

Figura 3.5: Camadas lógicas de um sistema embarcado em um microcontrolador utilizando um interpretador.

Conclusão

Neste capítulo, foram apresentados os dispositivos IoT, seus usos, tipos e modelos de execução de aplicações (quando em microcontroladores ou em microprocessadores). É extremamente importante identificar todos os requisitos que a aplicação (sistema embarcado) executará nos dispositivos IoT – é por meio desses requisitos que se definirá o melhor dispositivo a ser utilizado, dado seus limites e capacidade de processamento.

O próximo tópico a ser discutido é o modo como esses dispositivos se comunicam com o exterior, seja essa comunicação feita com outros dispositivos em uma rede ou com sistemas distribuídos. No próximo capítulo, são abordados os protocolos e modelos de comunicação utilizados na comunicação de dispositivos IoT,

arquiteturas de rede e acesso à internet comumente encontrados em soluções IoT.

Capítulo 4
Redes e protocolos

Dada a ampla variedade de tecnologias empregadas para conectar dispositivos em uma solução IoT, algumas dessas tecnologias são muito familiares ao nosso dia a dia, como o wi-fi e o Bluetooth. Outras tecnologias são menos conhecidas no nosso jargão tecnológico, mas ainda muito adotadas no mundo de IoT e constantemente utilizadas ao nosso redor (sem termos o menor conhecimento sobre elas).

Em IoT, as tecnologias utilizadas para comunicação entre dispositivos podem variar muito porque sua comunicação depende do tipo de dispositivo adotado, do local onde estão, com quais dispositivos se comunicam e o tipo de informação que compartilham. Conhecer as tecnologias e os padrões de comunicação de dispositivos em uma solução IoT é um passo muito importante para construções de melhores soluções.

Como veremos ao longo deste capítulo, não existe um melhor protocolo ou tecnologia que seja essencial, comum e global para todos os dispositivos. A tecnologia de comunicação ideal depende das necessidades da solução sendo construída (FALUDI, 2021).

A diversidade de cenários (que faz o mercado de IoT ser tão rico) é o fator que torna o desenho de soluções tão complexo e importante. Dificilmente uma solução será completamente igual a outra. Restrições de velocidade de tráfego de dados, volume de tráfego de dados, distância entre dispositivos, tamanho dos dispositivos e consumo de energia são alguns dos fatores que devem ser levados em consideração e que guiam a adoção de tecnologias em um desenho de solução. Não apenas considerando os termos técnicos, também é preciso que sejam avaliados os seus custos de implementação – e, assim, definir a sua viabilidade.

Cenários demasiadamente complexos, se não bem planejados (e utilizando as tecnologias corretas), tornam sua implementação inviável ou extremamente cara. O custo é um fator crítico e decisivo. Quando o custo de um projeto é maior do que o seu benefício, então a sua viabilidade se torna discutível.

É preciso dedicar tempo à pesquisa, descoberta e análise dos recursos tecnológicos disponíveis para que se construa uma arquitetura de solução que vá ao encontro das necessidades dos usuários dentro de um custo viável.

Ao longo deste capítulo, veremos que tecnologias de comunicação utilizadas em IoT possuem um conjunto comum de características, como alcance, velocidade de tráfego, consumo de energia, segurança, padrão de comunicação (por exemplo, *gateway*, acesso direto, *mesh*), meios de tráfego da informação (por exemplo, sistema cabeado, ondas de rádio) etc. Quando tais características são colocadas em conjunto, uma tecnologia de comunicação é formada e mais adequada a um conjunto de dispositivos, com o objetivo de melhor resolver um problema do mercado (por exemplo, comunicação a longa distância e economia de bateria).

Serão apresentados detalhes de múltiplas tecnologias de comunicação utilizadas em soluções de IoT, além de cenários de uso comuns a essas tecnologias. Também veremos protocolos e padrões de comunicação utilizados para a troca de mensagens entre dispositivos e/ou entre dispositivos e sistemas distribuídos.

4.1 Comunicação entre dispositivos e acesso à internet

Todo dispositivo IoT precisa comunicar-se de alguma forma. O conceito-chave por trás de IoT é a troca de informação: sem ela, não existe IoT.

Existem três tipos de comportamento adotados por dispositivos IoT:

1. Dispositivos que apenas enviam informações – executam alguma tarefa localmente e a enviam para outro dispositivo ou sistema distribuídos;
2. Dispositivos que apenas recebem informações – recebem informações (ou comandos) externas e reagem a elas;
3. Dispositivos que enviam e recebem informações: recebem informações (ou comandos) externas, reagem a elas e enviam resultados a sistemas distribuídos.

Apesar de conter “internet” em seu nome, nem todo dispositivo em uma solução IoT possui acesso direto à internet. O acesso à internet é caro, pois requer maior consumo de energia, dispositivos mais caros e complexos, proteções de segurança mais robustas e maior manutenção.

Neste capítulo, os dispositivos são separados em dois tipos:

- **Dispositivos diretamente conectados à internet:** esses dispositivos possuem acesso à internet e são responsáveis por enviar e receber dados de serviços remotos. Por terem acesso à internet, são mais complexos e custosos de serem mantidos. São possíveis portas de entrada para ataques de usuários maliciosos e requerem maior proteção;

- **Dispositivos indiretamente conectados à internet:** dispositivos que não possuem acesso à internet. São responsáveis por executar um trabalho e comunicar-se na mesma rede com outros dispositivos, que podem ou não ter acesso direto à internet.

Para distribuir as responsabilidades entre os dispositivos conectados a uma rede, economizar recursos computacionais e de energia, são adotados padrões de comunicação. Tais padrões de comunicação são amplamente utilizados no mercado e frequentemente observados ao nosso redor.

A seguir, abordaremos os padrões de comunicação contidos neste livro.

Gateway

Quando um dispositivo é incapaz de acessar a internet diretamente (pois não suporta o protocolo TCP/IP), este ainda pode enviar suas mensagens a um outro dispositivo que possua acesso à internet: um *gateway*. O *gateway* vai processar essa informação e direcioná-la para outro sistema distribuído através da internet.

Gateways são muito comuns e importantes para soluções IoT: eles suportam requisições de múltiplos dispositivos e tornam soluções IoT mais acessíveis, pois permitem que a maioria dos dispositivos em uma solução continue simples, barata e funcionando por baterias.

A figura a seguir apresenta uma rede onde múltiplos dispositivos conectam-se a um *gateway* que recebe as requisições e as propaga para outros sistemas conectados à internet.

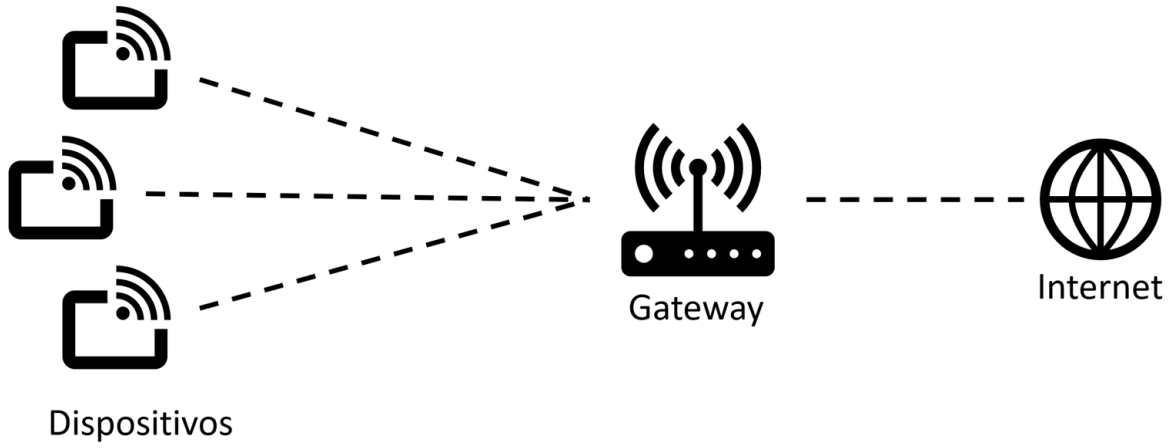

Figura 4.1: Estrutura de rede com múltiplos dispositivos conectados a um único gateway conectado à internet.

Gateways também são utilizados para a tradução de protocolos: dispositivos conectam-se ao *gateway* utilizando um protocolo;

gateway recebe as mensagens e as converte em um formato diferente, próprio para um protocolo mais comum à internet, como HTTP e MQTT.

Mesh

Mesh é uma topologia de rede onde os nós (dispositivos na rede) conectam-se uns aos outros, formando uma malha de conexões. Os nós em uma rede *mesh* conectam-se ao maior número possível de nós adjacentes, cooperando uns com os outros para que os dados sejam eficientemente trafegados – autorreconfigurando-se para otimizar sua malha de conexões, caso necessário.

Um dispositivo com acesso à internet funciona como nó principal (*gateway* de requisições), enquanto os demais nós (dispositivos sem acesso à internet) comunicam-se entre si, ampliando a malha e o alcance da rede. Dispositivos mais distantes do nó principal utilizam dispositivos adjacentes como intermediários para comunicarem-se com o nó principal.

A figura seguinte apresenta uma rede *mesh* onde múltiplos dispositivos estão conectados entre si. Os dispositivos mais distantes da internet propagam suas mensagens para os dispositivos mais próximos. O dispositivo conectado à internet funciona como *gateway*, recebendo requisições dos dispositivos mais próximos, e as propaga para servidores na rede ou serviços na nuvem.

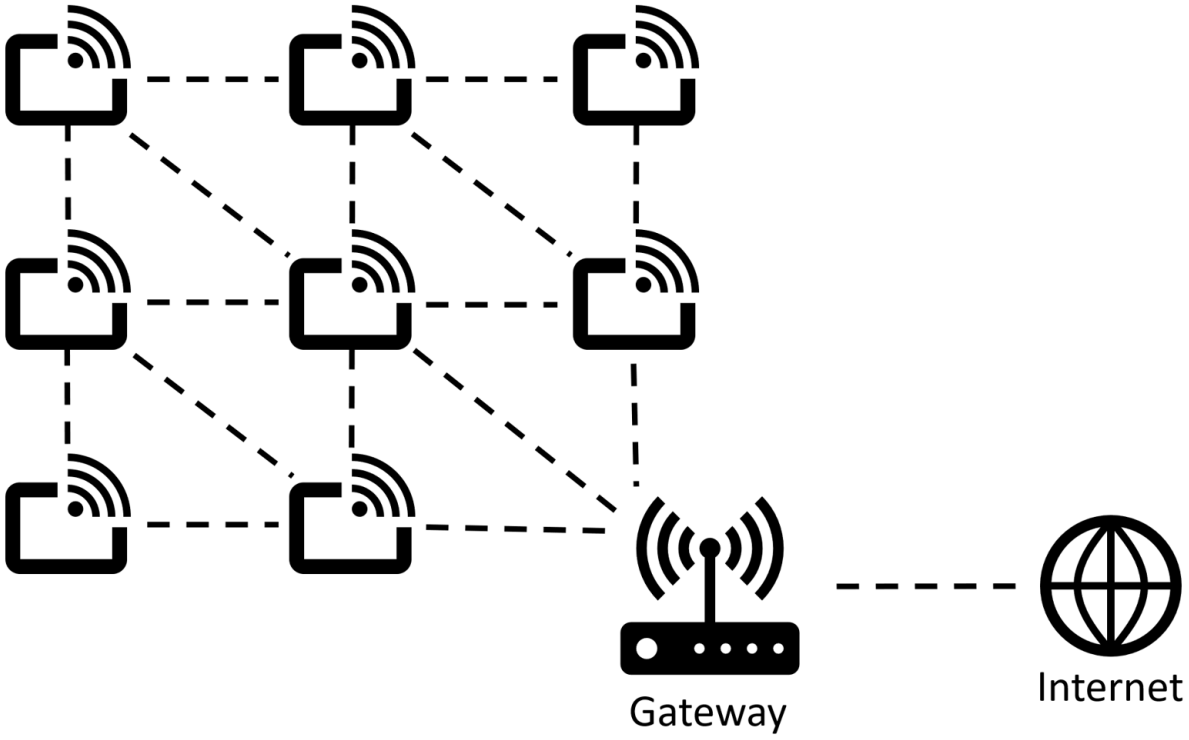

Figura 4.2: Estrutura de uma rede mesh conectada à internet.

Acesso direto

Este modelo é bastante simples e aplicável apenas em dispositivos que suportam o protocolo TCP/IP. Seu acesso é direto à internet, sem a necessidade de dispositivos intermediários. Neste modelo de acesso, os dispositivos são responsáveis por estabelecer a conexão com a internet e emitir mensagens por si. É recomendado em cenários em que não existem restrições de energia e o volume de dados trafegado é alto.

A figura seguinte demonstra quatro diferentes dispositivos conectando-se diretamente à internet.

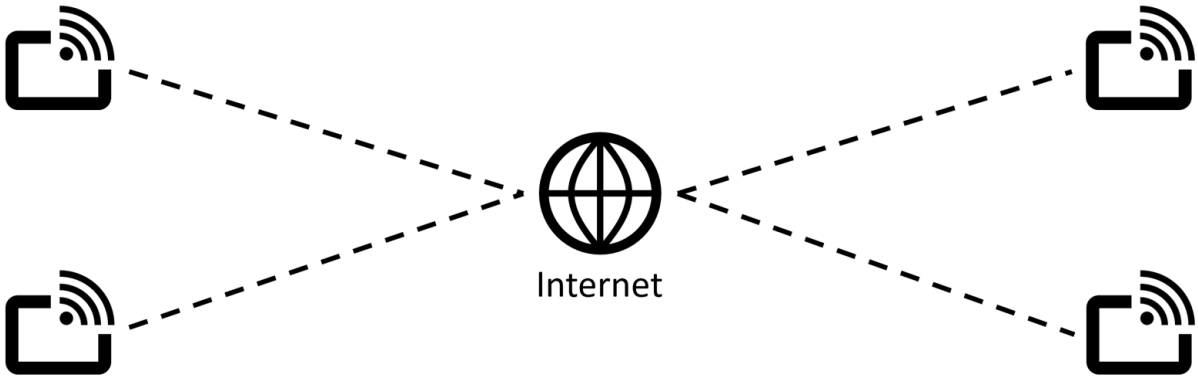

Figura 4.3: Múltiplos dispositivos com acesso direto à internet.

4.2 Dispositivos e tipos de redes

Dispositivos podem trafegar dados de diferentes formas, seja essa comunicação com outros dispositivos ou servidores. Nesta seção são apresentados os tipos de rede mais comuns utilizados em soluções IoT, assim como seus cenários de uso. Os tipos de rede nesta seção são divididos em duas características: sistemas não baseados em IP (*Internet Protocol*) e sistemas baseados em IP.

Este capítulo trata de diversas alternativas de redes sem fio para comunicação de dispositivos IoT e sabemos que o alcance dessas redes pode ser o fator determinante de sua escolha. Procuramos trazer uma referência do alcance dessas redes com base na documentação oficial de cada tipo de rede e protocolo, mas é crucial saber que o alcance de redes sem fio pode variar bruscamente em função do ambiente onde os dispositivos estão instalados.

Alguns dos exemplos de fatores que influenciam o alcance de redes sem fio são:

1. **Frequência da rede utilizada**: frequências mais altas (superiores a 70 Mhz, no espectro de VHF – *Very High Frequency*) tendem a sofrer menos com interferências do meio

ambiente. Todos os tipos de rede citados neste capítulo atuam em frequências muito superiores a este limite, na casa de 1 GHZ ou mais (no espectro de UHF – *Ultra High Frequency*).

2. **Potência do transmissor *versus* sensibilidade do receptor**: na maioria dos casos, a potência máxima a ser utilizada é determinada pelo tipo de rede escolhida, pela legislação de cada país para aquela banda de frequências e, por fim, pelo consumo esperado do dispositivo IoT. O uso de receptores de rádio mais sensíveis, presentes nos dispositivos de mais qualidade, também ajuda no desempenho total da solução. Para referência, a potência de transmissores é dada em watts (W) e a sensibilidade dos receptores é dada em decibéis (db).

3. **Visada direta *versus* visada indireta**: refere-se à existência de obstáculos entre o transmissor e o receptor. Esses obstáculos podem ser naturais, como uma montanha, ou obstáculos criados pelo homem, como prédios. Cada obstáculo vai atenuar o sinal gerado pelo transmissor e, se existirem muitos obstáculos, a barreira pode se tornar intransponível.

4. **Uso de antenas externas e direcionais**: uma maneira de aumentar o alcance é o uso de antenas externas, que devem ser específicas para a frequência utilizada. Embora seja mais complexo o seu uso, antenas direcionais são o meio mais eficiente de transmitir um sinal de rádio entre dois pontos por concentrar a energia em uma direção única.

5. **Ruído eletromagnético**: ambientes hostis são aqueles onde existem grandes motores, geradores, fiação elétrica exposta, indutores potentes (como os encontrados em máquinas de ressonância magnética) e onde fagulhas/faíscas possam ser geradas (como nos motores de combustão interna). Como a maioria das fábricas, hospitais, automóveis e ambientes industriais possuem múltiplos geradores de ruído eletromagnético, o funcionamento de dispositivos IoT geralmente é restrito ao uso de conexão cabeada.

Por esses motivos, considere todas as informações relacionadas ao alcance de redes sem fio como mera referência.

Sistemas não baseados em IP

Sistemas sem fio não baseados em IP são otimizados para reduzir o consumo de energia. Um termo bastante comum quando se discutem redes sem fio é WPAN (*Wireless Personal Area Network*, ou, em português, Área Pessoal de Rede Sem Fio), que se trata de um modelo de rede dedicada à comunicação de dispositivos centralizados em uma área comum.

O significado do termo “pessoal” para WPAN parte da premissa de que essa rede atenderá apenas a um único indivíduo – por exemplo, uma pessoa que possua um smartphone e um relógio inteligente que se comunicam entre si; uma pessoa que, ao entrar em um ambiente, acende as luzes utilizando um comando de voz; um sistema de segurança que avisa o morador de uma casa quando uma porta se abre.

Nesta seção, são apresentadas tecnologias de rede sem fio não baseadas em IP implementadas em soluções IoT.

Bluetooth

Bluetooth é um protocolo popular no mercado de tecnologia, cuja principal característica é o baixo consumo de energia e a troca de informação entre dispositivos próximos fisicamente. Amplamente utilizado em smartphones, teclados, mouses e outros periféricos, o Bluetooth também é utilizado extensivamente em IoT, suportando soluções para monitores de saúde, sistemas de alarme, controles remotos, sistemas de localização *indoor* e outros.

O protocolo Bluetooth possui duas versões (e que não se comunicam entre si): Bluetooth clássico e BLE (*Bluetooth Low Energy*, em português, Bluetooth de Baixa Energia), que foi projetada para dispositivos que precisam consumir baixas quantidades de energia. O padrão adotado por redes Bluetooth

utiliza ondas de rádio UHF (*Ultra High Frequency*) entre 2.402 e 2.480 GHz.

Redes Bluetooth foram projetadas para suportar uma ampla variedade de distâncias entre dois dispositivos, de 5 metros até 2 quilômetros de distância. Uma série de fatores podem interferir no alcance do sinal Bluetooth, tais como: espectro de rede, PHY (*Physical layer*, em português, camada física), sensibilidade do receptor, energia gasta na transmissão, poder de recepção da antena e perda de sinal durante a transmissão (BLUETOOTH SIG, 2022). Esses fatores, além de influenciarem no alcance das redes, também interferem na taxa de transmissão dos dados.

A camada física de uma rede sem fio define o seu esquema de modularização e outras técnicas utilizadas para enviar dados por meio de uma frequência de rádio. Redes Bluetooth suportam múltiplas opções de camadas físicas, cada uma com diferentes características que definem o alcance e a taxa de transmissão. A tabela a seguir apresenta taxas de transmissão suportadas por redes Bluetooth clássicas e *Bluetooth Low Energy* usando diferentes camadas físicas.

Tabela: Camadas físicas utilizadas por redes Bluetooth e suas respectivas taxas de transmissão.

Rede Bluetooth	Camada física	Taxa de transmissão
Bluetooth clássico	EDR PHY (8DPSK)	3 Mb/s
Bluetooth clássico	EDR PHY (π/4 DQPSK)	2 Mb/s
Bluetooth clássico	BR PHY (GFSK)	1 Mb/s
Bluetooth Low Energy	LE 2M PHY	2 Mb/s

Rede Bluetooth	Camada física	Taxa de transmissão
Bluetooth Low Energy	LE 1M PHY	1 Mb/s
Bluetooth Low Energy	LE Coded PHY (S=2)	500 Kb/s
Bluetooth Low Energy	LE Coded PHY (S=8)	125 Kb/s

Existem pré-requisitos para que um dispositivo possa ser compatível com Bluetooth; para tanto, o dispositivo precisa possuir um microprocessador que suporte Bluetooth, assim como um outro dispositivo (que também suporte Bluetooth) para que seja feita a comunicação (ARELLANO, 2019).

Bluetooth e BLE são populares e amplamente utilizados em IoT. Como Bluetooth não implementa o protocolo TCP/IP, é necessário que outro dispositivo – como um telefone celular – seja utilizado como dispositivo intermediário para o acesso à internet (FALUDI, 2021).

Exemplo de uso - Smartwatches

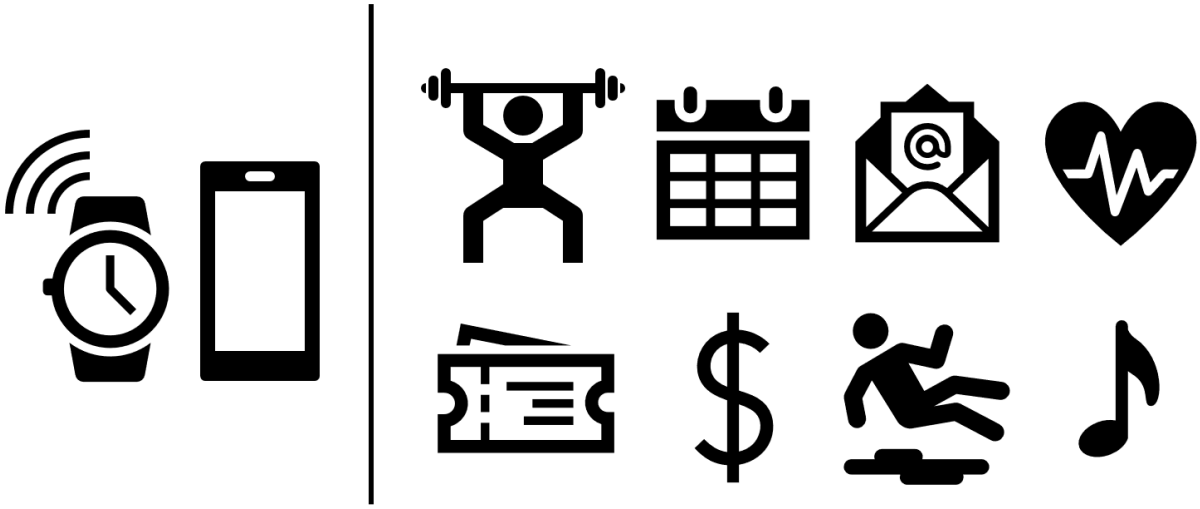

Figura 4.4: Smartwatches e alguns de seus modos de utilização: controle de calendário, pagamento, identificação de quedas, monitoramento cardíaco etc.

O poder dos smartphones é único: eles mudaram o modo como as pessoas veem e consomem o mundo ao seu redor. Ao mesmo tempo, permitiram que dispositivos que até então não possuíam qualquer tipo de conectividade “tomassem carona” em seu sucesso e reescrevessem seu uso – o que é o caso dos smartwatches (em português, relógios inteligentes).

Smartwatches conectam-se aos smartphones via Bluetooth e os utilizam como *gateway* de acesso à internet, provendo inúmeras novas possibilidades de uso a seus usuários. Por exemplo, com smartwatches que acompanham seus movimentos, usuários podem acompanhar o progresso de suas atividades físicas, ritmo cardíaco, tempo gasto em exercícios, tempo gasto em pé, tempo em descanso, distâncias percorridas e mais.

Usuários podem também realizar tarefas simples, como visualizar e responder e-mails e mensagens de texto, interagir com eventos em calendários, pagar compras (utilizando NFC, *Near-Field Communication*), atender e responder chamadas telefônicas.

Fabricantes de software têm aproveitado a mobilidade dos smartwatches aliada à sua conectividade (por estarem conectados

via Bluetooth a um dispositivo conectado à internet) e desenvolvido aplicações focadas em facilitar a vida de seus usuários, permitindo que eles controlem dispositivos remotamente. Por exemplo:

- Abrir e fechar portões eletrônicos de garagem;
- Ativar sistemas de segurança domiciliar;
- Acender lâmpadas;
- Controlar fechaduras remotamente;
- Gerenciar o funcionamento de eletrodomésticos (por exemplo, fogões, churrasqueiras, aspiradores de pó).

Smartwatches têm se tornado cada vez mais indispensáveis na área de saúde. Os profissionais da área de saúde recomendam o uso destes dispositivos para que pacientes possam acompanhar seus batimentos cardíacos, qualidade do sono e atividades físicas. Pacientes também podem optar por compartilhar suas informações com seus provedores de saúde para conseguirem melhores avaliações de como podem melhorar seu estado físico. Pesquisas demonstram que 57% das pessoas passaram a se exercitar mais após adquirirem um smartwatch (NORTHWEST PRIMARY CARE, 2021). No entanto, smartwatches ainda vão além e permitem:

- Detectar quedas e acionar ajuda;
- Detectar irregularidades no batimento cardíaco;
- Identificar níveis sonoros que possam afetar a audição de seus usuários;
- Auxiliar na contagem de calorias e auxiliar em um planejamento melhor das refeições.

Zigbee

IEEE 802.15.4 é um protocolo de rede sem fio que opera em três frequências de rádio: 868 MHz, 915 MHz e 2400 MHz. O protocolo IEEE 802.15.4 não é uma tecnologia por si só; ele é, na verdade, a base para que outras tecnologias de rede existam (ELECTRONICS NOTES, 2021) – por isso a sua importância.

O protocolo IEEE 802.15.4 define os fundamentos para redes WPAN de baixo consumo de energia, baixa velocidade e comunicação ubíqua entre dispositivos. Originalmente desenvolvido para suportar comunicação em rádio com alcance mínimo de 10 metros, com taxa de transferência de 250 kbit/s. Taxas mais baixas de transferência de dados podem ser adotadas, caso o consumo de energia tenha prioridade.

Um dos objetivos do protocolo IEEE 802.15.4 é reduzir os custos de produção e venda de dispositivos utilizados em redes WPAN, visto que seus requisitos técnicos podem ser alcançados por hardwares relativamente mais simples.

Zigbee é uma rede WPAN baseada no protocolo IEEE 802.15.4. O objetivo do Zigbee é ser mais barato e simples do que outras soluções WPAN (como o Bluetooth) e mais genérico do que outras soluções de rede sem fio (como o wi-fi). Zigbee opera em frequência de banda de 2.4 GHz (CSA, 2022).

O Zigbee é utilizado em soluções que envolvam dispositivos:

- com baixo consumo de energia – baterias que duram de 1 a 2 anos sem qualquer necessidade de recarga;
- com baixa mobilidade – dispositivos que não são movidos frequentemente;
- com baixa transferência de dados – limite de 250 Kbits/s;
- com alcance entre 75 e 300 metros para ambientes internos e de até 300 metros para ambientes externos;
- que sejam usados em pequenos e médios espaços supervisionados, onde se há total controle sobre o ambiente – por exemplo, o interior de casas, escritórios, quartos de hospitais e pequenos comércios;
- com comunicação segura – redes Zigbee são encriptadas por chaves simétricas de 128 bytes.

Dispositivos dentro de uma mesma rede Zigbee podem transmitir dados em longa distância através do tráfego de informações por

padrão de comunicação *mesh*, no qual dispositivos próximos funcionam como dispositivos intermediários na comunicação com dispositivos mais distantes. Redes Zigbee suportam até 65 mil dispositivos em uma mesma rede (FRENZEL, 2012).

Zigbee é atualmente mantido pela Connectivity Standards Alliance (originalmente criado e mantido pela Zigbee Alliance em 2002), que visa desenvolver uma plataforma aberta (não proprietária) para comunicação de dispositivos IoT. Esse grupo também certifica produtos para garantir a interoperabilidade entre dispositivos, além de promover o uso da tecnologia.

Exemplo de uso – Escritório inteligente

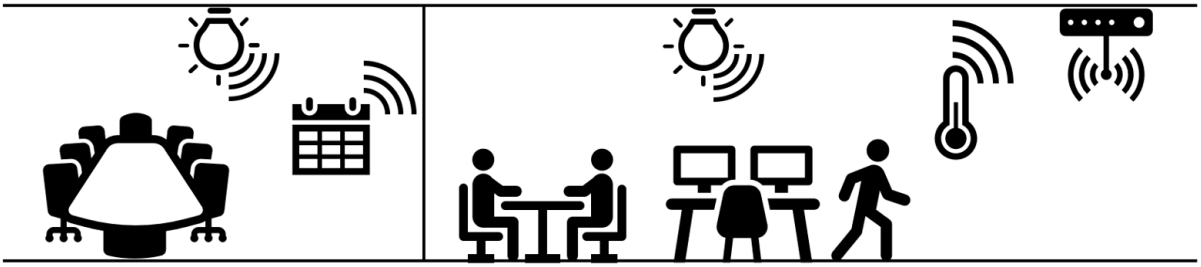

Figura 4.5: Em um escritório inteligente, dispositivos se comunicam com um gateway Zigbee para troca de mensagens e comandos.

Ambientes de trabalho estão constantemente evoluindo e incorporando IoT para aperfeiçoar a experiência de trabalho de seus funcionários, reduzir custos e utilizar recursos de forma eficiente.

Empresas têm adotado soluções utilizando Zigbee por ser uma solução eficiente para pequenos e médios espaços, onde se há controle sobre o ambiente e baixa (ou quase nula) mobilidade dos dispositivos utilizados.

Zigbee tem sido utilizado para auxiliar em soluções como:

- Monitores externos às salas de reunião que informam os funcionários sobre o calendário de reservas daquele dia;

- Sensores que contabilizam o número de pessoas em uma sala de reunião e gerenciam o sistema de refrigeração para adequar a temperatura ambiente;
- Sensores de movimento que acendem as luzes automaticamente ou as apagam na ausência de movimento;
- Cortinas eletrônicas que se ajustam automaticamente devido à posição do sol ou devido ao aquecimento externo, afetando a temperatura interna do escritório.

Z-Wave

Z-Wave é um modelo de rede sem fio que utiliza o padrão de comunicação *mesh* para o tráfego de informações, suportando até 232 dispositivos em uma mesma rede (FRENZEL, 2012). Z-Wave foi construído para controlar e monitorar dispositivos conectados em casas e pequenos ambientes comerciais.

Este tipo de rede sem fio suporta a comunicação entre dispositivos com até 30 metros de distância indoor e 100 metros de distância outdoor, com taxas de transferência de até 100 Kb/s. Z-Wave opera em ondas de rádio abaixo de 1 GHz (Z-WAVE ALLIANCE, 2022).

Assim como Zigbee, Z-Wave é um protocolo muito utilizado em automações residenciais, porque é otimizado para baixo consumo de energia, comunicações de baixa largura de banda e para uma comunicação segura e direta entre dispositivos. Z-Wave não suporta o protocolo TCP/IP, o que exige que seu acesso à internet seja feito por meio de um *gateway* (FALUDI, 2021).

Z-Wave é mantido pelo Z-Wave Alliance, um grupo composto por líderes da indústria e que se dedica ao desenvolvimento e extensão do Z-Wave.

Em setembro de 2020, Z-Wave Alliance anunciou uma nova especificação chamada Z-Wave Long Range. Z-Wave Long Range estende as capacidades do Z-Wave, permitindo:

- Maior cobertura – conectividade a quilômetros de distância;

- Aumento de escalabilidade – suporta até 4 mil dispositivos em uma mesma rede;
- Otimização do consumo de bateria;
- Retrocompatibilidade com dispositivos que suportem Z-Wave;
- Aumento da taxa de transferência – velocidade mínima de 100 Kb/s.

Como o Z-Wave Long Range ainda é uma especificação nova (em relação à data de lançamento desta obra), ainda não existem números que indiquem o sucesso de seu uso ou relevância no mercado de IoT.

Exemplo de uso – Sistema de segurança domiciliar

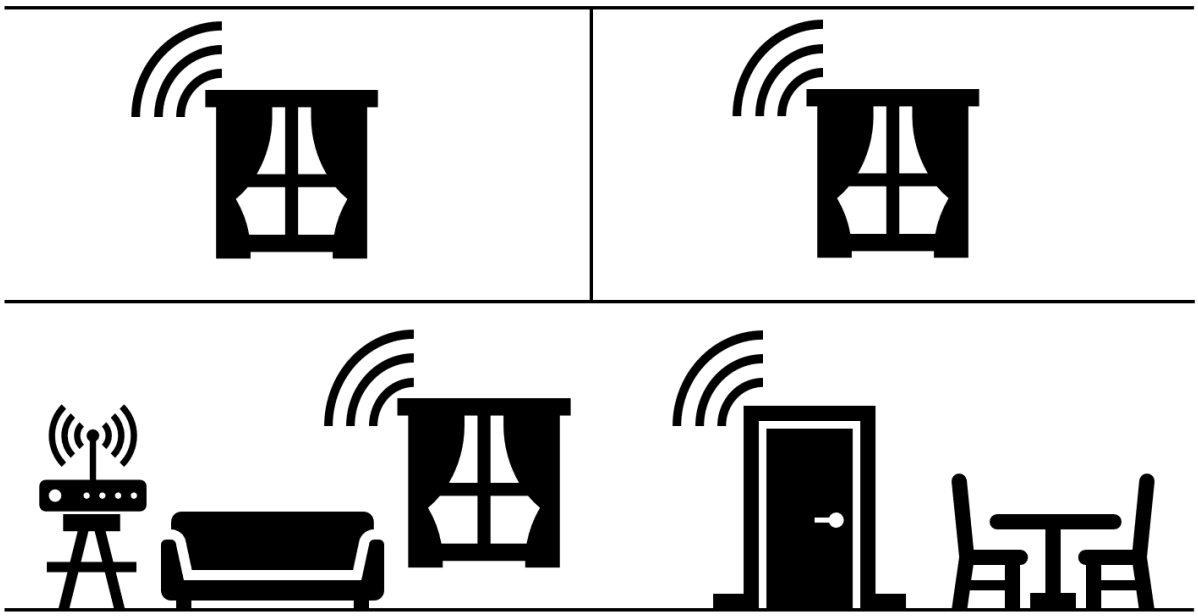

Figura 4.6: Em uma casa inteligente, dispositivos se comunicam com um gateway Z-Wave para troca de mensagens e comandos.

Para proteger seu patrimônio, uma família decidiu por instalar um sistema de segurança em sua casa. Preferindo por uma solução de mercado pronta, foi escolhido um kit de segurança composto por dois sensores de movimento, quatro sensores de contato (para portas e janelas), um amplificador de sinal e uma central de segurança, que gerencia o estado de todos estes sensores.

A central de segurança conecta-se à internet via TCP/IP e funciona como *gateway* Z-Wave para todos os outros dispositivos do kit. Quando uma porta é aberta, o estado do sensor de contato é modificado, exigindo que este se comunique com a central de segurança para notificar sua mudança de estado. Se o sistema de segurança estiver ativo, então a central de segurança entra em alerta, comunicando-se externamente com sistemas distribuídos e notificando os proprietários que houve uma mudança no estado de um dos sensores de contato conectado a uma das portas da residência.

Para evitar que dispositivos posicionados nos pontos mais extremos da residência percam sua conectividade, um amplificador de sinal também faz parte do kit de segurança. Esse amplificador de sinal funciona como um repetidor de sinal Z-Wave, que recebe uma mensagem de um dispositivo mais distante da central de segurança e o amplifica para que não exista perda na propagação de mensagens.

LoRa

Quando discutimos sobre redes de computadores utilizando sinais de rádio, é quase impossível não lembrar por um instante das emissoras de rádio que transmitem seus sinais por longas distâncias. Para que esses sinais de rádio sejam fortes o suficiente para trafegar por muitos quilômetros, as antenas utilizadas pelas emissoras de rádio precisam ser grandes e consumir muita energia. Em IoT, energia é, muitas vezes, um recurso escasso, o que torna a comunicação a longo alcance difícil de ser implementada. Por conta disso, o mercado de redes e comunicações tem investido em LPWAN (*Low-Power Wide-Area Network*): redes de comunicação de longo alcance que consomem pouca energia.

Dentre as redes LPWAN, LoRa é a de maior destaque. LoRa (*Long Range*, em português, longo alcance) é uma tecnologia de rede ideal para cenários particulares de IoT graças à sua combinação única de características: comunicação de longo alcance sem fio e

baixo consumo de energia. LoRa opera em uma região de frequências não regulamentada, portanto, é de uso gratuito. A frequência varia em todo o mundo, mas no Brasil é de 915Mhz.

Dispositivos conectados a uma rede LoRa podem comunicar-se com múltiplos receptores a distâncias de até 10 quilômetros. E o uso de energia necessário para o tráfego de pacotes LoRa é baixo (próximo a 25 miliwatts; VOITOVYCH, 2019). Dispositivos podem ter seu consumo de energia otimizado, utilizando a mesma bateria por anos sem necessidade de recarga.

Redes LoRa transferem dados em uma frequência de 0.3 kb/s a 50 kb/s (LORA ALLIANCE, 2022). Essa é uma taxa de transferência baixa quando comparada a outros tipos de redes. O objetivo de redes LoRa não é trafegar grandes volumes de dados a longa distância em alta velocidade, mas suprir cenários de comunicação com localizações remotas ou de difícil acesso – onde pequenos pacotes de dados podem ser utilizados para ativar sistemas, monitorar, emitir atualizações periódicas de estado ou alarmar situações críticas.

LoRa segue os padrões estabelecidos pelo protocolo LoRaWAN (*Long Range Wide-Area Network*). LoRaWAN é um padrão de comunicação aberto mantido pela LoRa Alliance, uma associação sem fins lucrativos que visa aperfeiçoar e manter o protocolo LoRaWAN.

Dispositivos precisam autenticar-se antes de se conectarem à rede. Isso torna as redes LoRa mais seguras ao restringir seu acesso e não permitir que qualquer dispositivo possa se conectar e interceptar pacotes sendo trafegados.

Para que uma rede LoRa seja construída, é preciso que exista ao menos um dispositivo cliente e um *gateway*. O dispositivo cliente emite ou recebe pacotes LoRa do *gateway*, enquanto o *gateway* é responsável por manter a conexão com os dispositivos clientes e conectar-se à internet.

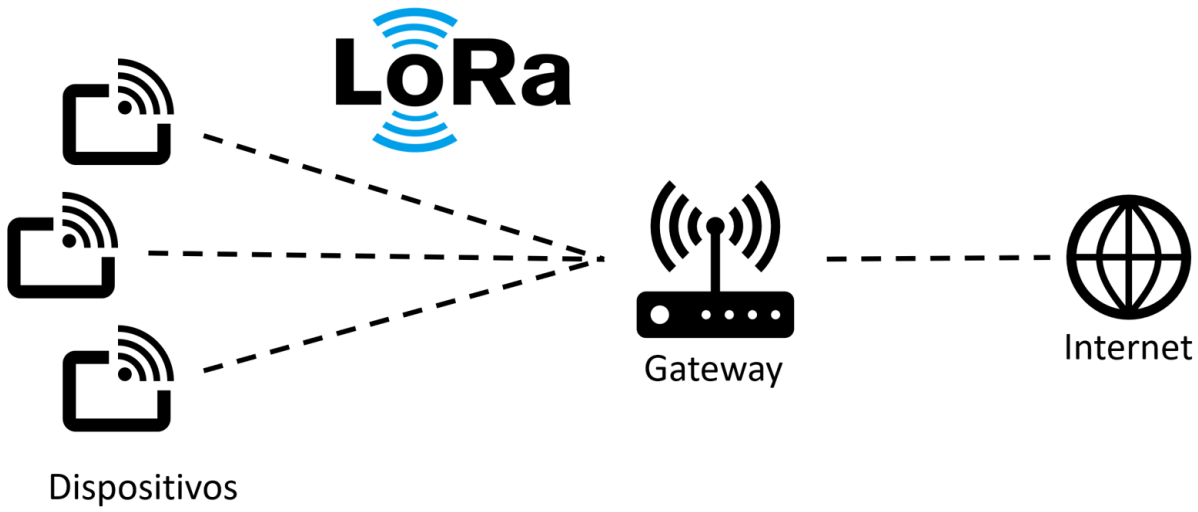

Figura 4.7: Representação de uma rede LoRa contendo múltiplos dispositivos clientes comunicando com um único gateway LoRa.

Exemplo de uso – Montanhas inteligentes

Figura 4.8: No alto de uma montanha, dispositivos conectado a uma rede LoRa enviam telemetria para seus receptores na base dos alpinistas.

Em diversos pontos de uma cordilheira de montanhas, são instalados pontos de análise com dispositivos que registram tremores de terra, temperatura, umidade e calculam as condições para ocorrência de avalanches.

Tais informações são enviadas via LoRa para múltiplas bases de alpinistas que se localizam aos pés da cordilheira. As equipes que coordenam as bases de alpinistas ficam atentas às possibilidades de avalanche e coordenam as equipes, mantendo-as seguras.

LoRa é ideal nesse cenário por diversos motivos: habilidade em comunicar-se a longas distâncias e com múltiplos pontos ao mesmo tempo; independência de serviços de terceiros e por sua alta disponibilidade.

Exemplo de uso - Fazenda conectada

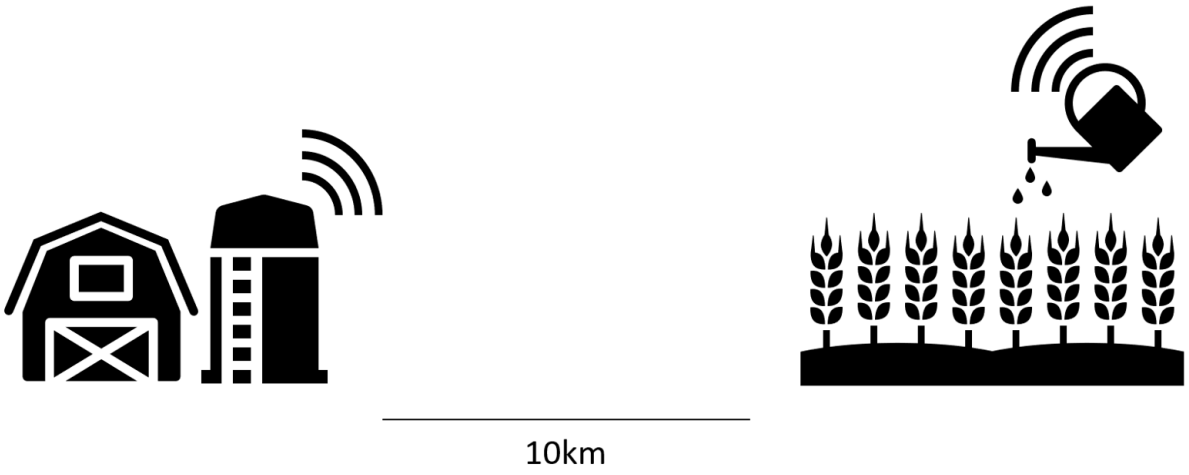

Figura 4.9: Dispositivos IoT espalhados por uma fazenda conectam-se à sua central de controle via LoRa, enviando mensagens e recebendo comandos.

Em uma fazenda de soja, um dos sistemas de irrigação eletrônica comunica-se com a central (a quase 2km de distância) utilizando LoRa, informando o nível de umidade de um dos setores de plantio.

A central da fazenda recebe este sinal e identifica que o índice de umidade deste setor está abaixo do recomendado para aquela época do ano. A central da fazenda emite um sinal para a unidade de irrigação requisitando que aquele setor seja irrigado. O sistema de irrigação recebe o sinal e começa a irrigar a plantação.

A comunicação via LoRa é ideal neste cenário devido à distância e ao baixo tráfego de informação entre o sistema de irrigação e a central de controle da fazenda. Apenas poucas trocas de mensagens durante o dia já são suficientes para a manutenção de cada setor da fazenda.

RFID

Radio-Frequency Identification, também conhecido como RFID, corresponde ao uso de campos eletromagnéticos para identificação e rastreio de etiquetas (*tags*) acopladas a objetos.

RFID não é um tipo de rede de computadores, mas é um recurso muito utilizado em soluções IoT para a identificação de objetos dentro de um perímetro: um leitor de sinais RFID pode identificar a presença de uma etiqueta RFID desde poucos centímetros até vários metros de distância.

RFID é comumente utilizada em:

- Pagamento de meios de transporte, como metrôs, trens, ônibus e pedágios em estradas;
- Meio de autenticação física para abertura de portas e cancelas (como em crachás e chaveiros eletrônicos);
- Autenticação em dois fatores (*two-factor authentication*).
- Controle de inventários, rastreio de ativos e controle de cadeia de suprimentos;
- Prevenção de falsificações (como em cartuchos de impressoras).

Um sistema de RFID consiste em ao menos uma etiqueta RFID e uma base RFID, que funciona como leitora de sinais. As etiquetas RFID possuem um circuito integrado e uma antena, e podem ser ativas ou passivas.

- Passivas: não possuem alimentação de energia elétrica própria, portanto só emitem sinal de resposta quando próximas da leitora de sinal RFID, pois se utilizam da energia de rádio transmitida pela leitora para funcionar.
- Ativas: possuem alimentação de energia elétrica própria e conseguem emitir sinal RFID sem estarem próximas à leitora de sinais, o que permite a comunicação a distâncias maiores em relação ao modelo passivo.

Etiquetas RFID passivas são mais baratas e menores, por não terem bateria. Também podem ser utilizadas por períodos mais longos de tempo, já que não precisam ter suas baterias recarregadas. Em contrapartida, seu uso exige a proximidade das leitoras.

Sistemas baseados em IP

Redes TCP/IP tornaram-se o modelo tradicional de comunicação entre sistemas e computadores. Apesar de consumirem mais energia do que os modelos de redes não baseados em IP, ainda são amplamente utilizadas em IoT.

Nessa seção, são apresentadas tecnologias de rede baseadas em IP implementadas em soluções IoT.

Wi-Fi

Wi-fi corresponde a uma família de tecnologia de redes sem fio, baseadas no protocolo IEEE 802.11, que utilizam diferentes frequências de rádio para transmissão de informações entre dispositivos.

O alcance de uma rede wi-fi pode variar de 20 até 150 metros, dependendo das barreiras existentes no local onde a rede estiver instalada. A sua velocidade de tráfego é bastante alta, podendo atingir até 9.6 Gb por segundo (Wi-Fi 6, suportando protocolo 802.11ax), permitindo a transferência rápida de grandes volumes de dados. Entretanto, por serem mais complexas, conexões wi-fi acarretam dispositivos que consomem mais energia durante o tráfego de informações.

Por suportar o protocolo TCP/IP nativamente, dispositivos com wi-fi abstraem a complexidade do acesso à internet naturalmente.

Para que um dispositivo seja compatível com wi-fi, é preciso que este possua um microcontrolador e um firmware que gerencie a integridade e a segurança das credenciais de acesso a redes. Tal necessidade por um firmware se dá pelo fato de que redes wi-fi são uma das possíveis portas de entrada para invasões de usuários maliciosos (ARELLANO, 2019).

Algumas das motivações para se utilizar wi-fi em uma solução IoT:

- Simplicidade de uso – conexões wi-fi são muito populares e fáceis de serem criadas. Soluções de hardware disponíveis no mercado oferecem processos de configuração simples e intuitivos que tornam a criação de redes wi-fi trivial;
- Disponibilidade de equipamentos compatíveis – existem muitas soluções de roteadores e repetidores wi-fi. Equipamentos compatíveis com as necessidades de sua solução e com preços acessíveis podem ser encontrados facilmente;
- Velocidade – em um cenário em que altos volumes de dados (por exemplo, vídeos, imagens em alta resolução) precisam ser transmitidos em alta velocidade;
- Segurança – para a transmissão em segurança de dados sigilosos, fi-fi provê múltiplos protocolos de segurança, como WEP, WPA, WPA2 e WPE2.

Para se construir uma rede Wi-Fi, é preciso que exista um dispositivo de rede que funcione como *access point* (em português, ponto de acesso) e que os demais dispositivos tenham a capacidade de se comunicar com esse *access point* utilizando uma das frequências de rádio suportadas pelo protocolo IEEE 802.11.

Em IoT, redes wi-fi são utilizadas em cenários em que:

- Seja necessário o alto tráfego de informações (por exemplo, streaming de vídeo, alto volume de telemetria);
- Não existem restrições de energia;
- A comunicação entre dispositivos é feita em um raio de 20 até 150 metros de distância (considerando possíveis barreiras físicas).

Exemplo de uso - Prédio online

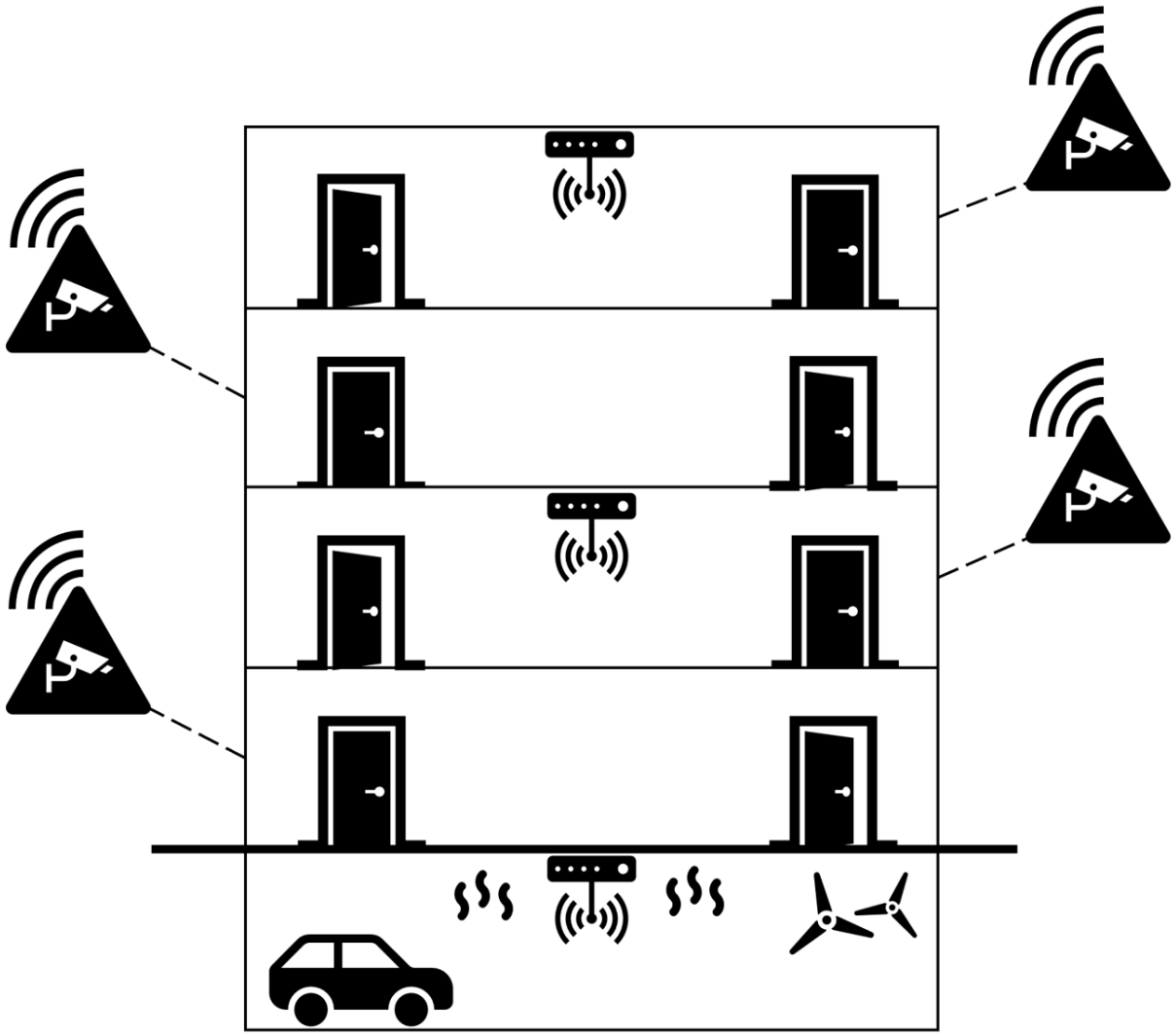

Figura 4.10: Em um prédio, dispositivos de segurança conectam-se a uma rede central via wi-fi.

Para garantir a segurança e o conforto dos moradores de um prédio de apartamentos, foram instalados diferentes sensores e dispositivos por todos os andares – todos conectados via wi-fi, com ao menos um repetidor de sinal por andar para que a qualidade de conexão não seja perdida.

Em todos os andares e ao redor do andar térreo, foram instaladas câmeras que se conectam ao sistema de segurança via wi-fi. As

câmeras monitoram 24h por dia o movimento dos moradores e possíveis pessoas não autorizadas.

Sensores de CO_2 foram instalados na garagem, ativando o sistema de ventilação quando os índices de CO_2 no ar estiverem próximos aos limites recomendados.

O controle de acesso via biometria também está conectado à rede wi-fi do prédio; nesse sistema são registradas as entradas dos moradores, assim como validada sua autenticação.

Para todos esses dispositivos, wi-fi é a solução ideal, por permitir streaming de vídeo, comunicação entre dispositivos e tráfego de dados em alta velocidade, além de ser uma solução barata e muito conhecida no mercado.

Ethernet

Dispositivos com conexões cabeadas (*ethernet*) utilizam o protocolo TCP/IP para se comunicarem com um servidor em uma rede ou uma aplicação na nuvem via internet. Conexões cabeadas são mais rápidas e estáveis, mas frequentemente é muito caro ou fisicamente impossível utilizar uma comunicação cabeada em longas distâncias (FALUDI, 2021).

Neste tipo de rede, cabos *ethernet* serão utilizados para conectar dispositivos a roteadores que proveem acesso à internet, estabelecendo uma simples conexão LAN (*Local Area Network*) tal qual estamos habituados.

A velocidade de conexão pode variar devido a restrições impostas pelo administrador da rede, ao tipo do cabo de rede, a limitações dos dispositivos de rede etc. Mas, em geral, sob condições normais, espera-se que uma conexão *ethernet* utilizando um cabo de rede de categoria 6 (Cat6) possa atingir até 1 GB por segundo.

Vantagens do uso de redes *ethernet*:

- **Estabilidade** – conexões *ethernet* são muito estáveis para transmissão de dados. Um cabo de *ethernet* possui maior resistência a interferências, danos físicos e outros fatores externos que interferem na comunicação entre dispositivos;
- **Custo** – baixo custo de implantação devido à sua popularidade no mercado de tecnologia;
- **Alta velocidade** – quando comparadas a outros tipos de rede nas quais é preciso gastar mais recursos para se obter maior velocidade, redes *ethernet* já suportam alta velocidade de tráfego de dados por padrão;
- **Simplicidade** – facilidade em adicionar novos dispositivos à rede.

Exemplo de uso - Fábrica inteligente

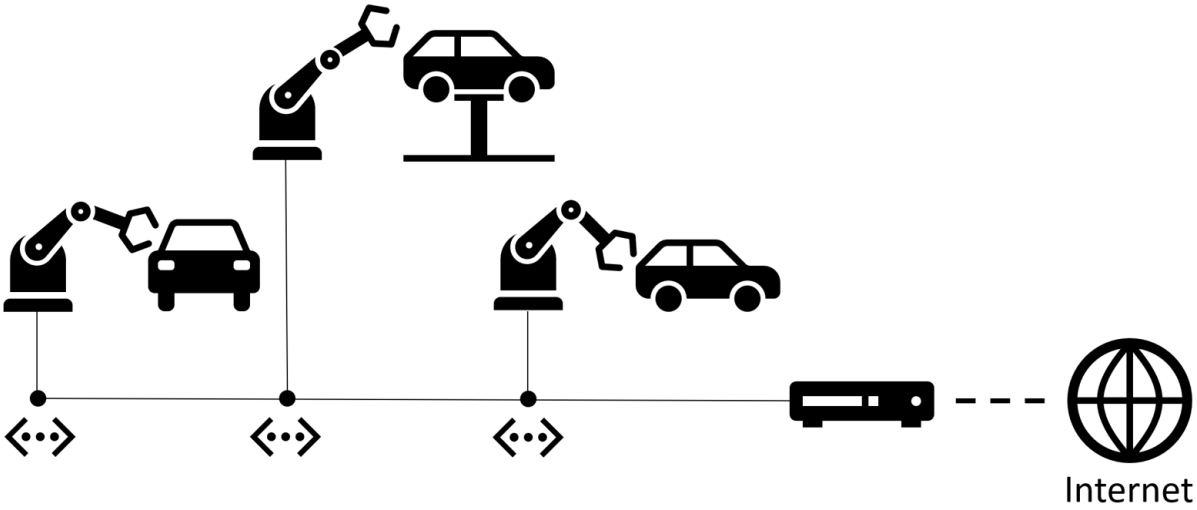

Figura 4.11: Robôs conectados à rede de uma fábrica enviando telemetria via internet.

Em uma fábrica de carros, braços robóticos executam diferentes funções: montagem de peças, soldagem, pintura, instalação de vidros, movimentação de veículos entre esteiras etc.

Cada braço robótico está conectado a um sistema que gerencia seu estado, as tarefas executadas ao longo do dia e a precisão de seus movimentos, a fim de identificar a necessidade de manutenção ou novos ajustes e evitar erros que possam danificar os veículos em produção.

Apesar de a fábrica ser um ambiente supervisionado e controlado, ruídos eletromagnéticos gerados pelas máquinas podem tornar proibitivo o uso de redes sem fio. Para suportar o volume de dados trafegados, uma rede *ethernet* foi provisionada para comunicação dos braços robóticos com seu sistema de gerenciamento. Esta rede, por ser cabeada, não sofre com os ruídos eletromagnéticos gerados pelos robôs e permite uma comunicação estável de alto volume de dados e com alta velocidade.

Cellular IoT

Implantar uma rede wi-fi que cubra a área de uma grande cidade e que suporte a alta transmissão de dados para todos seus usuários pode ser muito custoso ou tecnicamente inviável. Desta forma, foi necessária a busca por novas soluções.

Redes de celular tornaram-se uma solução bastante viável, pois conseguem acessar distâncias muito maiores do que wi-fi e podem facilmente suportar dispositivos IoT, além de não exigirem um roteador que estabeleça a conexão com a internet (o que torna sua adoção mais simples).

Cellular IoT é o termo utilizado para classificar dispositivos IoT conectados à internet por meio de redes de celulares.

Por muito tempo, dispositivos conectados a redes de celulares consumiram muita energia para manter-se conectados com qualidade. Foi então necessário o desenvolvimento de novas tecnologias que melhor se adequassem às limitações de dispositivos IoT, permitindo que dispositivos operem por longos períodos, transmitindo pacotes de informação de tempos em tempos sem a necessidade de recarregar suas baterias ou constante alimentação de energia.

Atualmente, a maioria dos dispositivos IoT conectados a redes de celulares utilizam LTE-M ou NB-IoT, ambas desenvolvidas por 3GPP (*3rd Generation Partnership Project*), um grupo de sete organizações de telecomunicações que trabalham no desenvolvimento de novos padrões e tecnologias na área.

LTE-M (*Long-Term Evolution for Machines*) é um tipo de rede de longo alcance de baixo consumo de energia. LTE-M oferece alta taxa de transmissão de dados, mobilidade e voz pela rede. Entretanto, é caro e não é aplicável em soluções que não utilizem todo seu potencial. Aplicações que utilizem alta transmissão de dados, como carros autônomos e serviços de emergência em tempo real, são aconselhados a utilizar LTE-M.

NB-IoT (*Narrowband Internet of Things*) é um tipo de rede otimizado para dispositivos que não se movem ou que requerem baixo tráfego de dados (acarretando menor consumo de energia). NB-IoT é uma opção mais barata do que LTE-M.

Outros protocolos, como LTE, 4G e 5G, também são utilizados no mercado. Estes exigem mais energia e são mais caros (por não serem otimizados para IoT), mas suportam maior tráfego de dados (por exemplo, vídeos e imagens em alta resolução) (FALUDI, 2021).

Exemplo de uso - Controle de tráfego

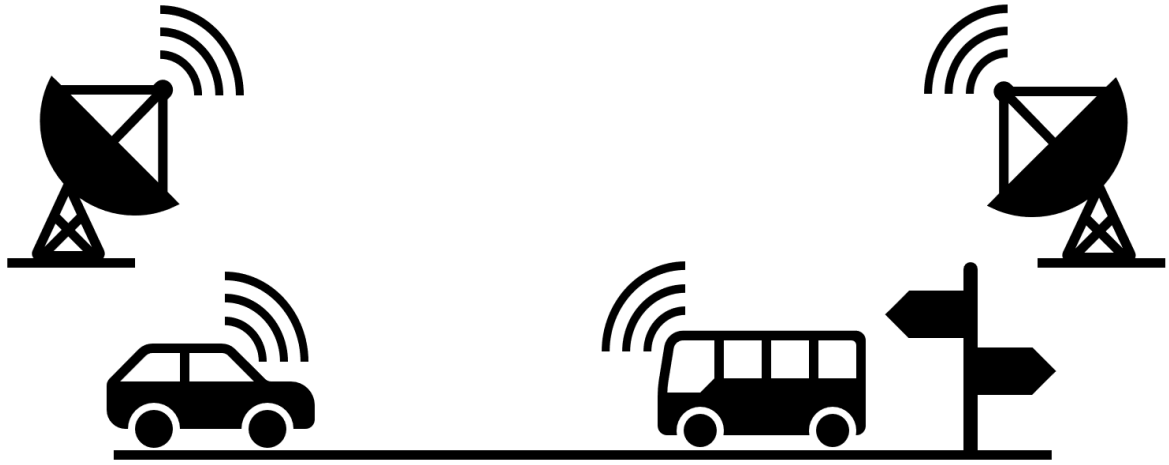

Figura 4.12: Veículos em um cidade conectam-se a um sistema central utilizando Cellular IoT.

Uma grande empresa possui sua matriz distribuída em 120 prédios diferentes em 5 cidades vizinhas. Para facilitar a movimentação de seus funcionários entre os prédios, uma frota de carros e micro-ônibus é ofertada.

Funcionários podem requisitar caronas entre os prédios por meio de um sistema online. Tal sistema identifica os veículos disponíveis mais próximos aos funcionários e os direciona ao seu auxílio.

Ao mesmo tempo, os funcionários requisitantes podem visualizar onde os veículos estão e o tempo estimado de sua chegada.

Com os veículos movimentando-se constantemente, suas posições são atualizadas no sistema frequentemente, além de notificar sua disponibilidade e checar por novas solicitações de carona.

Cada veículo está conectado à rede de celulares da região (via 3G, 4G ou 5G), emitindo dados pertinentes à posição e ao estado do veículo. Redes de celulares são ideais neste cenário, pois suportam a frequente comunicação de objetos em movimento, dados complementares à localização dos veículos e constante tráfego de informações.

Satélite

Satélites são utilizados para prover cobertura de rede em áreas remotas da Terra onde nem mesmo existe cobertura de sinal de celular. Soluções para *broadcasting* de informações e fornecimento de internet via satélite estão cada vez mais comuns no mercado e as empresas de telecomunicação, para melhor atender às necessidades de IoT, têm adaptado suas ofertas de mercado fornecendo acesso à internet via satélite nas regiões mais distintas do planeta.

Os dispositivos se conectam à internet via satélite por meio de BGAN (*Broadband Global Area Network*), uma rede global de satélites. Para conectar-se à BGAN é preciso que o dispositivo esteja conectado a um terminal (via USB, wi-fi ou *ethernet*). Terminais BGAN podem acessar a internet em qualquer lugar do mundo, basta existir acesso direto ao céu para que uma conexão seja estabelecida (operação que pode levar de um a dois minutos).

Devido ao seu preço alto, BGAN é utilizado em cenários críticos, como resposta a desastres, telemedicina, monitoramento remoto e uso militar.

Exemplo de uso - Sistema de emergência marítimo

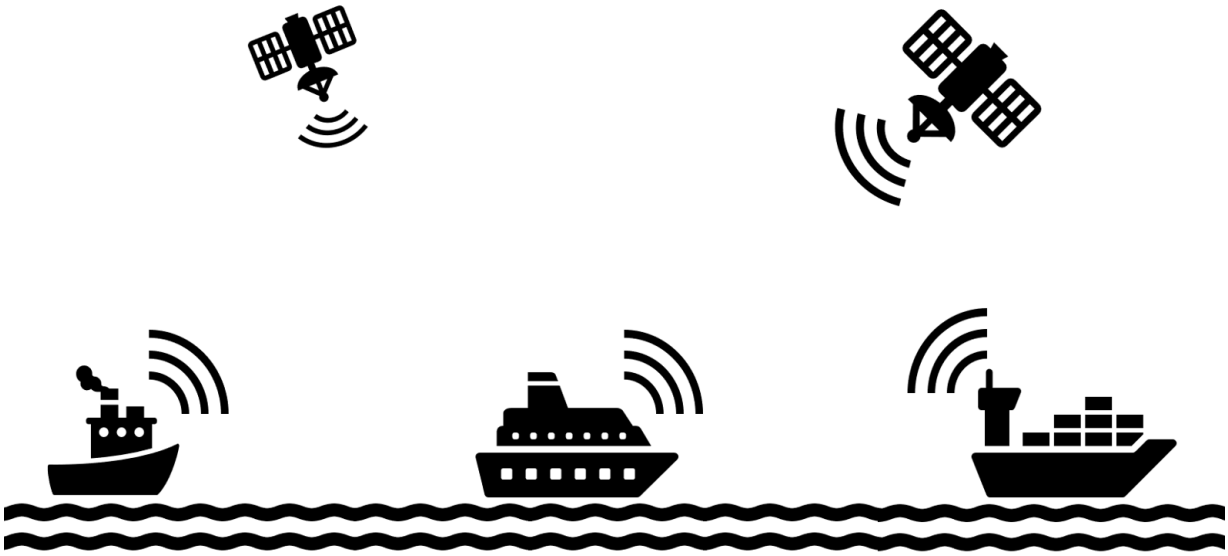

Figura 4.13: Embarcações espalhadas ao redor do mundo enviando telemetria via satélite.

Ao redor do mundo, um Sistema Eletrônico de Alerta e Emergência baseado em comunicação via satélite registra mensagens críticas de embarcações espalhadas pelos mares.

Embarcações enviam sinais com dados sobre mudanças inesperadas em suas condições, sinais de alerta ou pedidos de ajuda. Não importa onde estiverem ao redor do mundo, tais sinais serão recebidos pelos satélites e então enviados para estações em terra, onde equipes especializadas entrarão em ação.

4.3 Protocolos

Durante a definição do tipo de rede que será adotado por sua solução, também é necessário avaliar os protocolos de comunicação disponíveis. Protocolos de comunicação possuem limitações e modos de uso distintos, que podem afetar o modo como sua solução foi planejada.

Protocolos de comunicação são importantes pois permitem que dispositivos de diferentes fabricantes se comuniquem, além de estabelecerem práticas comuns para todos os dispositivos. A seguir são apresentados os protocolos de comunicação mais comuns em soluções IoT:

MQTT

MQTT (*Message Queue Telemetry Transport*) é um protocolo de enfileiramento de mensagens próprio para pequenos dispositivos que trafegam pequenos volumes de dados. É bastante utilizado em IoT e vem se tornando cada vez mais comum dada a sua flexibilidade, simplicidade, baixa sobrecarga de protocolo e adoção do padrão arquitetural *Publisher and Subscriber* (em português, publicador e assinante). Esse protocolo tornou-se tão essencial para IoT que o próprio mercado se adaptou a ele: hoje, todos os provedores de serviço de nuvem suportam MQTT.

O protocolo MQTT é regido pelo padrão MQTT 3, MQTT 3.1 e – mais recentemente – MQTT 5, cada um com suas particularidades. Como arquiteto ou arquiteta, cabe a você definir quais das funcionalidades suportadas por cada modelo são necessárias ao seu projeto.

Ao utilizar um serviço que suporte MQTT ou um *gateway* MQTT em sua rede, esse recurso é então chamado de *message broker*. A responsabilidade do *message broker* é definida como um conector entre publicadores de conteúdo e assinantes desse conteúdo. O conteúdo é organizado em tópicos, de forma que os publicadores podem definir exatamente para qual tópico vão enviar dados e os assinantes definem de quais tópicos desejam receber notificações.

MQTT trafega dados binários sobre o protocolo TCP/IP. Ele utiliza um modelo de comunicação onde o dispositivo sempre está conectado, o que otimiza o consumo de recursos do dispositivo, pois não se conecta e desconecta ao *message broker* toda vez que a transmissão de dados é necessária.

A qualidade de serviço (ou QoS, do inglês *Quality of Service*) também deve ser considerada. Ela representa a garantia de entrega de uma mensagem entre o MQTT *broker* e o dispositivo e pode variar entre os níveis QoS 0, QoS 1 ou QoS 2.

QoS 0 representa a menor das garantias em que as mensagens **serão entregues no máximo uma vez**. Ao usar QoS 0, não existe a confirmação de recebimento por parte do *broker* e isso faz com que o fluxo de mensagens seja muito rápido. QoS 0 tende a ser utilizado em cenários em que muitas mensagens são enviadas por dia e a perda de uma ou algumas dessas mensagens não é um problema para a aplicação.

QoS 1 é o modelo mais utilizado na indústria por apresentar um bom balanço entre a garantia de que a mensagem vai ser **entregue ao menos uma vez** e o tráfego adicional de mensagens para possibilitar essa garantia. Quando o dispositivo envia uma mensagem para a nuvem, este mantém uma cópia dessa mensagem até que receba a confirmação da nuvem de que a mensagem foi recebida. Caso a confirmação não seja recebida em um tempo determinado, o dispositivo pode fazer o reenvio. A quantidade de vezes que o dispositivo fará o reenvio depende do modelo de negócios e, após a mensagem atingir o número máximo de tentativas de reenvio, a mensagem é apagada.

Uma inconveniência do uso do modelo QoS 1 é que a nuvem pode receber mensagens duplicadas, situação que pode ocorrer quando a entrega da confirmação é atrasada e o dispositivo faz o reenvio da mensagem.

QoS 2 deve ser utilizado quando existe a necessidade de garantir que cada mensagem seja **entregue exatamente uma vez**. Neste modelo, além de o dispositivo ter que manter uma cópia local da mensagem, a nuvem também precisa fazer o mesmo para garantir que mensagens duplicadas sejam identificadas e eliminadas. Essas cópias são mantidas até que as confirmações da nuvem para o dispositivo e do dispositivo para a nuvem sejam recebidas.

Esse modelo representa o maior custo adicional em uso de memória do dispositivo e na nuvem, além de exigir várias mensagens adicionais de confirmação para cada mensagem efetivamente enviada.

A maioria dos provedores de serviço suporta os níveis de qualidade de serviço QoS 0 e QoS 1, mas poucos suportam QoS 2, por razões diversas como complicações no sistema, aspectos de segurança e garantia de serviço.

AMQP

AMQP (*Advanced Message Queueing Protocol*) é um protocolo de enfileiramento de mensagens que suporta o tráfego de grandes volumes de dados. Ao contrário do MQTT, o AMQP não é dedicado a pequenos dispositivos – este é um protocolo genérico de comunicação que pode ser utilizado por diferentes tipos de soluções além de IoT. AMQP suporta o padrão *Publisher and Subscriber*, assim como o padrão *Request and Response* (em português, requisição e resposta). Assim como MQTT, AMQP trafega dados binários sobre o protocolo TCP/IP.

CoAP

CoAP (*Constrained Application Protocol*) foi projetado para permitir a conexão simples e rápida entre dispositivos sem alto tráfego de dados. Este protocolo é geralmente utilizado na comunicação máquina a máquina e é muito utilizado como solução paliativa quando outros meios de comunicação falham. Por ser muito simples e leve, mesmo em situações extremas de conectividade limitada ou bastante congestionada, CoAP continua trabalhando.

CoAP já foi um protocolo de comunicação mais popular no mercado, mas hoje já perdeu espaço para o MQTT. Ele também trafega dados binários sobre o protocolo TCP/IP.

HTTP

HTTP (*Hypertext-Transfer Protocol*) é utilizado globalmente na comunicação entre sistemas - não apenas em IoT. Este é um protocolo que funciona exclusivamente com o padrão *Request and Response*.

REST (*Representational State Transfer*) tornou-se o formato mais utilizado na construção de novos sistemas. A utilização de REST é mais simples e intuitiva do que a de demais formatos, como SOAP (*Simple Object Access Protocol*) ou WSDL (*Web Services Description Language*).

MQTT é um protocolo mais leve do que a combinação de HTTP e REST. MQTT não exige que o cliente conecte e desconecte a cada transmissão de dados, enquanto que requisições HTTP o exigem.

Conclusão

Não existe uma única tecnologia de rede que funcione para todos os cenários de IoT. É preciso conhecer os requisitos técnicos da solução, analisar as tecnologias disponíveis no mercado e adotar aquela que melhor atende tais requisitos.

Redes e protocolos são críticos para que soluções IoT conectem-se com recursos do mundo exterior, tais como os recursos existentes na nuvem. No próximo capítulo, são apresentadas soluções de software mantidas por provedores de nuvem para auxiliar na criação de soluções IoT.

Capítulo 5
Serviços na nuvem para IoT

Segundo o National Institute of Standards and Technology (NIST), a computação na nuvem pode ser definida como um modelo para permitir o acesso ubíquo, conveniente e sob demanda para um *pool* compartilhado de recursos (como redes, servidores, armazenamento, aplicativos e serviços), que podem ser rapidamente provisionados e liberados com esforço reduzido de gerenciamento ou de interação com o provedor de serviços.

O conceito de nuvem é utilizado na grande maioria dos cenários de IoT, em que dispositivos remotos com baixa capacidade de processamento conectam-se a esses recursos computacionais remotos para o envio de dados (telemetria) e o recebimento de comandos.

Embora comum, esse cenário não é uma regra: dispositivos remotos podem ser complexos e com alta capacidade de processamento, assim como o uso de recursos na nuvem não é obrigatório. Muitas soluções IoT podem utilizar um servidor local (e outros recursos) dentro da rede de uma empresa, escritório, prédio, fazenda etc.

Cenário local versus cenário conectado

Chamamos de *cenário local* quando a solução IoT não precisa estar conectada à internet por já ter todos os recursos de que necessita localmente, sem precisar conectar-se com o mundo exterior, como em um *cenário conectado*. Como exemplo, tomemos o cenário de uma fazenda onde IoT é utilizada na otimização das tarefas do dia a dia.

Nesta fazenda, diversas plantações possuem sensores de umidade de solo, sensores de composição química do solo, sensores de chuva, detecção de gases e câmeras. Os silos e as estufas também possuem sensores de luminosidade e umidade. Todos esses

sensores enviam dados por uma rede proprietária via LoRa para a sede da fazenda, onde um sistema central armazena e avalia esses dados para comandar regadores automatizados, controlar a adição de fertilizantes, indicar onde o uso de pesticidas é necessário e prover de uma forma centralizada o progresso das plantações.

A imagem a seguir apresenta o fluxo de informação e comandos entre os componentes da fazenda e o sistema central, onde todas as informações são mantidas localmente sem necessidade de acesso externo.

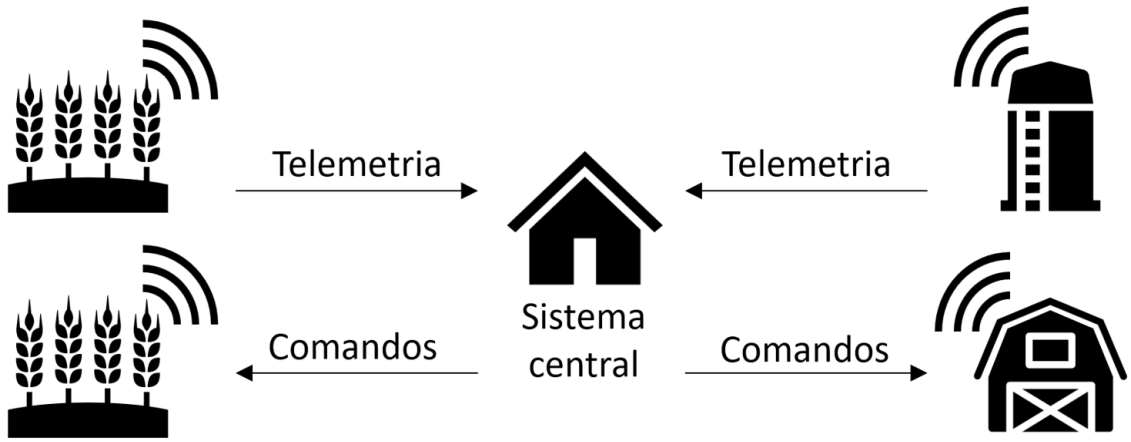

Figura 5.1: Fluxo de dados entre dispositivos de uma fazenda e seu sistema central.

A fazenda, nesse caso, é um ambiente fechado em que uma solução local pode ser o suficiente para seu objetivo de negócio. Prédios, fábricas, *campi* e outros espaços podem utilizar este mesmo modelo local. No entanto, o modelo em nuvem traz diversos benefícios que não podem ser ignorados, tais como escalabilidade, redundância, alta disponibilidade e velocidade de provisionamento de recursos.

5.1 Nuvem fora ou dentro da empresa?

Ao tomar a decisão de como alocar recursos de infraestrutura (localmente ou recorrendo a provedores externos), devemos considerar vários fatores, tais como:

Característica	Local	Nuvem
Custo	Toda a infraestrutura deve ser adquirida antes que se possa iniciar o uso do sistema. Capacidade não utilizada afeta diretamente o custo-benefício da solução, e aumentar a capacidade geralmente incorre em novas aquisições de hardware.	Todos os grandes provedores de nuvem (Amazon, Google e Microsoft) oferecem um modelo em que se paga apenas pelos recursos utilizados (*Pay as you go*) e que pode ser facilmente reduzido ou ampliado conforme a necessidade.
Manutenção	Como a infraestrutura é própria, toda a manutenção também é. Isso inclui servidores, equipamentos de rede, fornecimento de energia elétrica ininterrupta, sistema de redundância, ar-condicionado etc.	Ao contratar um provedor de nuvem, todas essas preocupações com manutenção são terceirizadas. A manutenção desses recursos é de responsabilidade do provedor de nuvem.

Característica	Local	Nuvem
Distribuição geográfica	Caso a solução de IoT necessite de servidores distribuídos geograficamente para minimizar a latência para os usuários finais, múltiplas cópias da infraestrutura são necessárias para cada distribuição. Nesse caso, a redundância e a comunicação entre os diversos servidores devem também ser consideradas.	Por padrão, os provedores de nuvem possuem presença global em diversos *data centers* ao redor do mundo. Dessa forma, criar e distribuir múltiplas infraestruturas semelhantes é trivial. O roteamento dos dados entre os *data centers* também é de responsabilidade dos provedores de nuvem.
Escalabilidade vertical	Aumentar a capacidade dos servidores da solução implica em uma intervenção física e aquisição de hardware.	Aumentar a capacidade dos servidores pode ser feito sem paradas inesperadas: ao mesmo tempo que um servidor recebe e processa requisições, seu substituto com maior capacidade de processamento está em processo de criação.

Característica	Local	Nuvem
Escalabilidade horizontal	Para aumentar a quantidade de servidores utilizados em uma solução, é necessário adquiri-los fisicamente. E quando eles não são mais utilizados, tendem a permanecer ociosos.	Inserir novos servidores em um ambiente na nuvem é uma tarefa trivial que pode ser automatizada: quando os sistemas gerenciadores percebem o aumento no consumo de recursos, novos servidores são adicionados. A remoção de servidores também ocorre no cenário oposto, quando recursos ficam ociosos e deixam de ser necessários.

Característica	Local	Nuvem
Segurança e certificações	No modelo local, aspectos de segurança, incluindo o acesso físico e lógico aos dispositivos IoT, servidores e seus recursos, são de responsabilidade própria.	Provedores de nuvem investem constantemente na proteção física e lógica de recursos. Times especialistas em segurança mantêm a infraestrutura segura, trabalhando constantemente na segurança de sua plataforma frente ao surgimento de novas possíveis ameaças. Certificações da indústria são mandatórias nestes serviços: setor público, área de saúde, militar e outros exigem a conformidade com normas ABNT e ISO de segurança.

Característica	Local	Nuvem
Soberania de dados	Quesito crucial para aplicações desenvolvidas para governos, a soberania de dados dita que as informações dessas aplicações não podem deixar o território nacional. Esse quesito pode ser facilmente atendido com uma solução local.	Dado que todos os grandes provedores de nuvem são empresas baseadas nos Estados Unidos, isso pode se tornar um empecilho para adoção de seus serviços quando a soberania nacional é uma preocupação. Nesse caso, deve-se procurar por empresas que possuam *data centers* locais e que tenham garantias de que estes dados não serão replicados para nenhuma outra região.

5.2 Infraestrutura, plataforma ou software como serviço?

Na tabela anterior comparamos as características de uma infraestrutura local com as de uma solução na nuvem. Os serviços na nuvem são construídos para suportar diferentes cenários de uso de seus clientes, e são divididos em três modelos: *Infrastructure as a Service*, *Platform as a Service* e *Software as a Service*. Cada um desses modelos possui um nível diferente de gestão de responsabilidades, em que um modelo possui maior responsabilidade sobre os recursos mantidos e, em outro, apenas se consomem tais recursos e toda manutenção é delegada ao provedor de nuvem.

Na figura a seguir, é apresentada a progressão de responsabilidades nos modelos mantidos pelos provedores de nuvem.

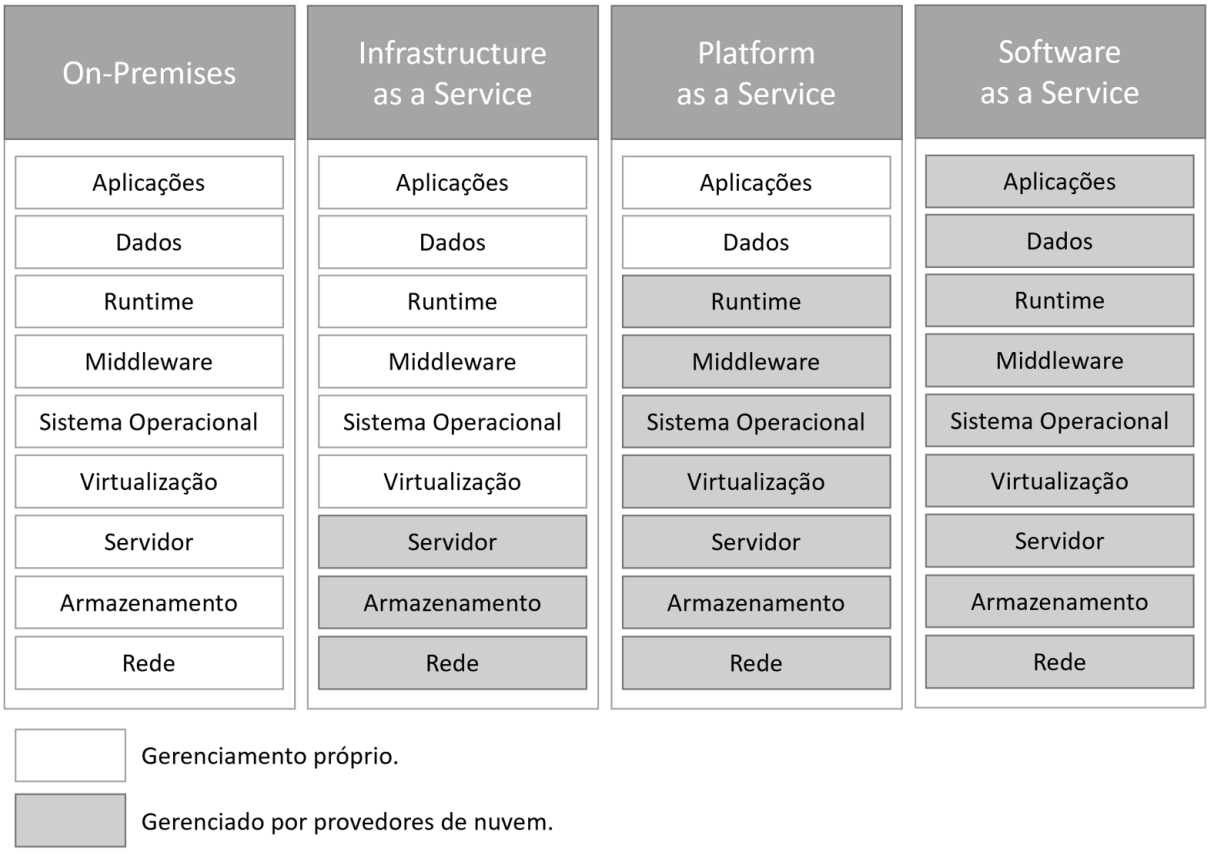

Figura 5.2: Progressão de responsabilidades nos modelos mantidos pelos provedores de nuvem versus soluções on-premises.

IaaS (Infrastructure as a Service) – Infraestrutura como serviço oferece recursos fundamentais de computação, armazenamento e rede sob demanda, pagos conforme o uso. Neste modelo, o ambiente e a infraestrutura física necessários para manter os servidores é de responsabilidade do provedor de nuvem (MICROSOFT, 2022a).

PaaS (Plataform as a Service) – Plataforma como serviço possui um nível de abstração de recursos computacionais maior do que IaaS, fornecendo serviços que enriquecem o desenvolvimento de soluções sem que usuários tenham de se responsabilizar em criar a

infraestrutura necessária desses serviços (MICROSOFT, 2022b). Este modelo fornece serviços cuja manutenção é transparente para o usuário final, tais como: serviços para ingestão e enfileiramento de dados; bancos de dados *serverless*; gestão de funções *serverless*; modelos online de treinamento e execução de inteligência artificial, entre outros.

SaaS (Software as a Service) – Software como serviço permite aos usuários se conectar e usar aplicativos baseados em nuvem. Exemplos comuns são e-mail, calendário e ferramentas do pacote Office, como o Microsoft Office 365 (MICROSOFT, 2022c). Este é o modelo de maior abstração de recursos computacionais, onde o usuário final precisa apenas realizar algumas configurações para utilizar o software.

Os próximos tópicos apresentam serviços disponíveis na nuvem e que são comumente utilizados em soluções IoT.

5.3 Message Broker

Message brokers são motores de processamento de eventos e comandos por meio da troca de mensagens. Muito populares na comunicação entre dispositivos e a nuvem, são baseados no padrão arquitetural conhecido como **PubSub** (publicação e assinatura/subscrição, do inglês *subscription*) de notificações de mensagens.

Ao consumir um *message broker*, a aplicação (*firmware* do dispositivo, também conhecido neste cenário como “produtor”) envia mensagens para um serviço remoto que provê recursos como roteamento, persistência, orquestração, governança e conversão de mensagens, além de suportar o streaming dos dados a todo destinatário (também conhecido como “consumidor”) que tenha subscrito a consumir o conteúdo submetido pelos dispositivos.

Mensagens são enfileiradas e processadas sequencialmente – garantir o processamento na ordem em que os dados foram recebidos é uma característica vital para os *message brokers*. Para garantir a rotatividade dos dados persistidos nos *messages brokers*, é comum encontrar soluções que associam a cada mensagem uma data de expiração, que determina o período de tempo em que uma mensagem estará disponível para ser consumida/processada.

Produtores e consumidores de dados comunicam-se com *message broker* utilizando tanto protocolos proprietários quanto protocolos padrões de mercado, como MQTT e HTTP. Provedores de nuvem fornecem serviços de *message brokers* que são escaláveis e suportam grande volume de mensagens, além de outras características importantes como alta disponibilidade, recuperação de desastres, redundância etc.

5.4 Identidade de dispositivos

Uma característica importante dos dispositivos de IoT é a sua identidade: cada dispositivo precisa ser unicamente identificado de forma a permitir sua gestão e de seus dados.

A gestão de dispositivos permite que ele seja individualmente desabilitado ou mesmo eliminado de uma rede sem afetar outros dispositivos. Dispositivos precisam ser desabilitados, ao menos temporariamente, quando detectadas anomalias em seu comportamento, como uma falha mecânica, elétrica ou até mesmo uma ação maliciosa de terceiros. Após um período de quarentena, dispositivos desabilitados podem ser readmitidos na rede.

Identidade única também se reflete em mais segurança por permitir que cada dispositivo use um método distinto de autenticação com credenciais únicas. Podemos ter um dispositivo usando um

certificado digital para se autenticar, enquanto outro utiliza uma solução mais simples como uma *connection string*.

Identidades duplicadas (quando dois ou mais dispositivos autenticam-se na rede usando as mesmas credenciais) devem ser evitadas, mas podem ocorrer por vários motivos, entre eles:

- Falha no provisionamento do dispositivo;
- Reúso indevido de credenciais por profissionais envolvidos na solução;
- Uso de credenciais duplicadas quando um dispositivo com uma sessão válida perde conexão e tenta se reconectar, gerando uma nova sessão. Caso a sessão anterior não seja eliminada, o número de sessões ativas pode ser maior do que o número de dispositivos;
- Ação maliciosa de criminosos que utilizam a identidade roubada de um dispositivo para atacar a solução.

A forma mais comum de tratar essa situação é não permitir a autenticação de dois dispositivos com a mesma credencial – caso em que o primeiro dispositivo será desconectado para que o segundo dispositivo se conecte.

5.5 Mensagens de telemetria

Telemetria corresponde ao fluxo de mensagens enviadas pelo dispositivo para a nuvem e é um dos cenários mais comuns em soluções IoT. A telemetria, também conhecida como *D2C* (do inglês *device to cloud*, "do dispositivo para a nuvem"), pode variar desde simples leituras de sensores de temperatura, umidade, pressão, rotação, até informações mais complexas como imagens e arquivos.

A maioria dos serviços impõe limites para o tamanho das mensagens de telemetria; então, é necessário verificar antecipadamente o tamanho esperado das mensagens de sua

aplicação para entender se o *payload* (conteúdo) deverá ser dividido entre múltiplas mensagens. Um exemplo é um provedor de serviço em que mensagens de telemetria não podem exceder 256 KB e mensagens que excedam esse tamanho deverão ser divididas em partes de, no máximo, 256 KB.

Além do tamanho da mensagem, outra limitação dos serviços pode ser a quantidade de mensagens suportada. Alguns deles oferecem diversos planos com variados limites de mensagens – por exemplo, planos básicos com limite de 400 mil mensagens por dia.

5.6 Mensagens de comando e controle

Mensagens de comando e controle são comumente utilizadas em soluções IoT. São mensagens da nuvem para os dispositivos com orientações de ações a serem tomadas, tais como: definir intervalo de telemetria, reiniciar o dispositivo, executar funções específicas. Também são conhecidas como *C2D* (do inglês *cloud to device*, "da nuvem para o dispositivo").

Os serviços podem ainda oferecer configurações adicionais, como o tempo limite para que o dispositivo confirme o recebimento de uma mensagem e a quantidade máxima de tentativas de entrega (quando o tempo de vida da mensagem expira, o provedor de serviço pode realizar tentativas adicionais). Os valores utilizados nessas configurações devem sempre estar alinhados com os requisitos de negócio da solução.

5.7 Dispositivos gêmeos

Dispositivos gêmeos (do inglês *device twins*), uma funcionalidade oferecida pelos provedores de serviço, correspondem à habilidade

de manter o sincronismo de propriedades entre os dispositivos e a nuvem. Propriedades são divididas entre desejadas e reportadas: propriedades desejadas são impostas pela nuvem; propriedades reportadas são aquelas informadas pelos dispositivos.

Como exemplo, pode-se citar a versão do *firmware* de um dispositivo: a nuvem (propriedade desejada) espera que o dispositivo esteja em uma versão específica, e o dispositivo (propriedade reportada) informa a sua versão atual. Se houver alguma diferença entre a propriedade desejada e a propriedade reportada, pode ser que o dispositivo precise atualizar-se.

Do ponto de vista do dispositivo, as propriedades desejadas não são modificáveis e sempre somente lidas (*read only*). As propriedades desejadas são sempre definidas pela nuvem e não podem ser alteradas pelos dispositivos.

Dispositivos gêmeos mantêm um documento para cada dispositivo conectado. Tal documento contém informações sobre o estado dos dispositivos, o que inclui metadados, configurações e outras condições. Ele também é armazenado localmente no dispositivo. Seu arquivo é dividido em seções que, embora variem em seu formato, contêm:

- **Identidade do dispositivo** – informações do dispositivo conforme armazenado no registro da nuvem permitem identificar unicamente o dispositivo e esta seção apenas existe na nuvem, não sendo enviada ao dispositivo;
- **Propriedades desejadas** – propriedades definidas pela nuvem que informam ao dispositivo o estado desejado para suas diversas propriedades. Exemplo: o dispositivo pode receber uma atualização do arquivo onde a propriedade desejada de temperatura seja 18 graus, ou a propriedade de intervalo de atualização de dados seja aumentada de uma vez a cada hora para uma vez a cada 10 minutos, entre outras. Esta seção

somente pode ser modificada pela nuvem, sendo apenas leitura para o dispositivo;

- **Propriedades reportadas** – propriedades definidas pelo dispositivo que informa à nuvem o estado reportado para suas diversas propriedades. Exemplo: o dispositivo pode enviar uma atualização do arquivo com propriedade reportada de status da última operação, ou a propriedade de versão do *firmware*, entre outras. Esta seção somente pode ser modificada pelo dispositivo, sendo apenas leitura para a nuvem;

A imagem a seguir apresenta a orquestração de propriedades por dispositivos gêmeos, na qual a nuvem é responsável por manter a identidade e as propriedades desejadas, enquanto os dispositivos mantêm as propriedades reportadas, reagindo a possíveis diferenças entre os estados desejados e reportados.

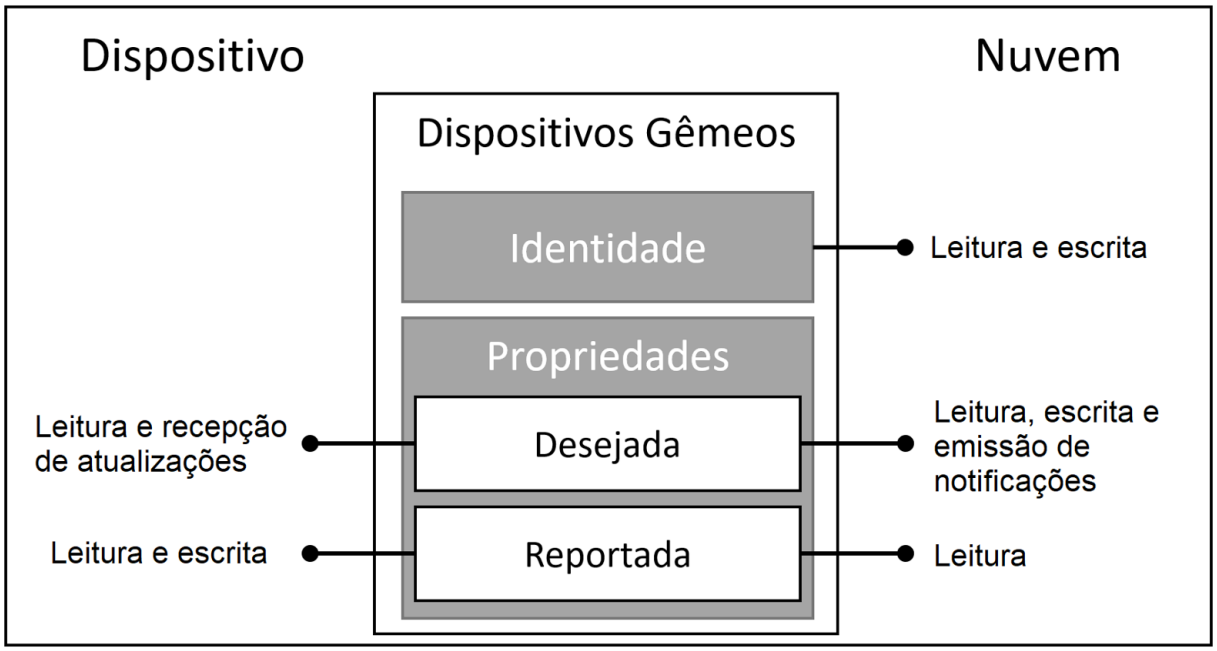

Figura 5.3: Orquestração de propriedades por dispositivos gêmeos (MICROSOFT, 2022d).

No modelo de dispositivos gêmeos, uma das grandes vantagens do uso de arquivos para a troca de informações é que este processo funciona mesmo para dispositivos que não estão conectados todo o

tempo: dado dispositivo que se conecta uma vez por dia (por semana, por mês etc.), quando conectado, receberá a última versão das propriedades desejadas e atuará de acordo com o definido. Da mesma forma, o dispositivo reportará suas propriedades, garantindo a sintonia entre ambos.

Para dispositivos conectados via redes tarifadas por dados (como rede de celular), enviar e receber o arquivo inteiro não é uma maneira otimizada de trabalhar; nesse caso, pode-se fazer apenas a troca das propriedades que sofreram alterações.

5.8 Provisionamento de dispositivos

Provisionamento de dispositivos (também conhecido como DPS, *Device Provisioning Service*) é o processo automatizado de criação de dispositivos na nuvem sob demanda. Uma vez registrado pelo fabricante na nuvem, ao se conectar pela primeira vez, o dispositivo tem suas credenciais de acesso criadas e enviadas para serem salvas localmente. Esse processo permite que o provisionamento de milhares de dispositivos ocorra sem intervenção humana (conhecido como *zero-touch provisioning*) e sem que as credenciais do dispositivo tenham de ser incluídas na aplicação de maneira explícita.

O provisionamento deve ser realizado em conexões seguras para que as credenciais utilizadas não sejam visíveis na rede em que elas trafegam.

Para ilustrar um cenário de uso de provisionamento de dispositivos, tomemos como exemplo uma empresa fabricante de portas de garagem inteligentes que, quando conectadas, permitem que seus usuários as abram ou fechem remotamente por um aplicativo de celular. Como esse produto é vendido por terceiros, não se possui controle de quando o dispositivo será conectado. Desta forma, o

provisionamento desse dispositivo deve ser feito em tempo real, quando (e se) o usuário desejar conectá-lo.

Nesse caso, o fabricante registra a lista de quais dispositivos podem ser provisionados posteriormente no serviço de nuvem. Essa lista é o que permite o fluxo de provisionamento, como demonstrado no diagrama de sequência a seguir:

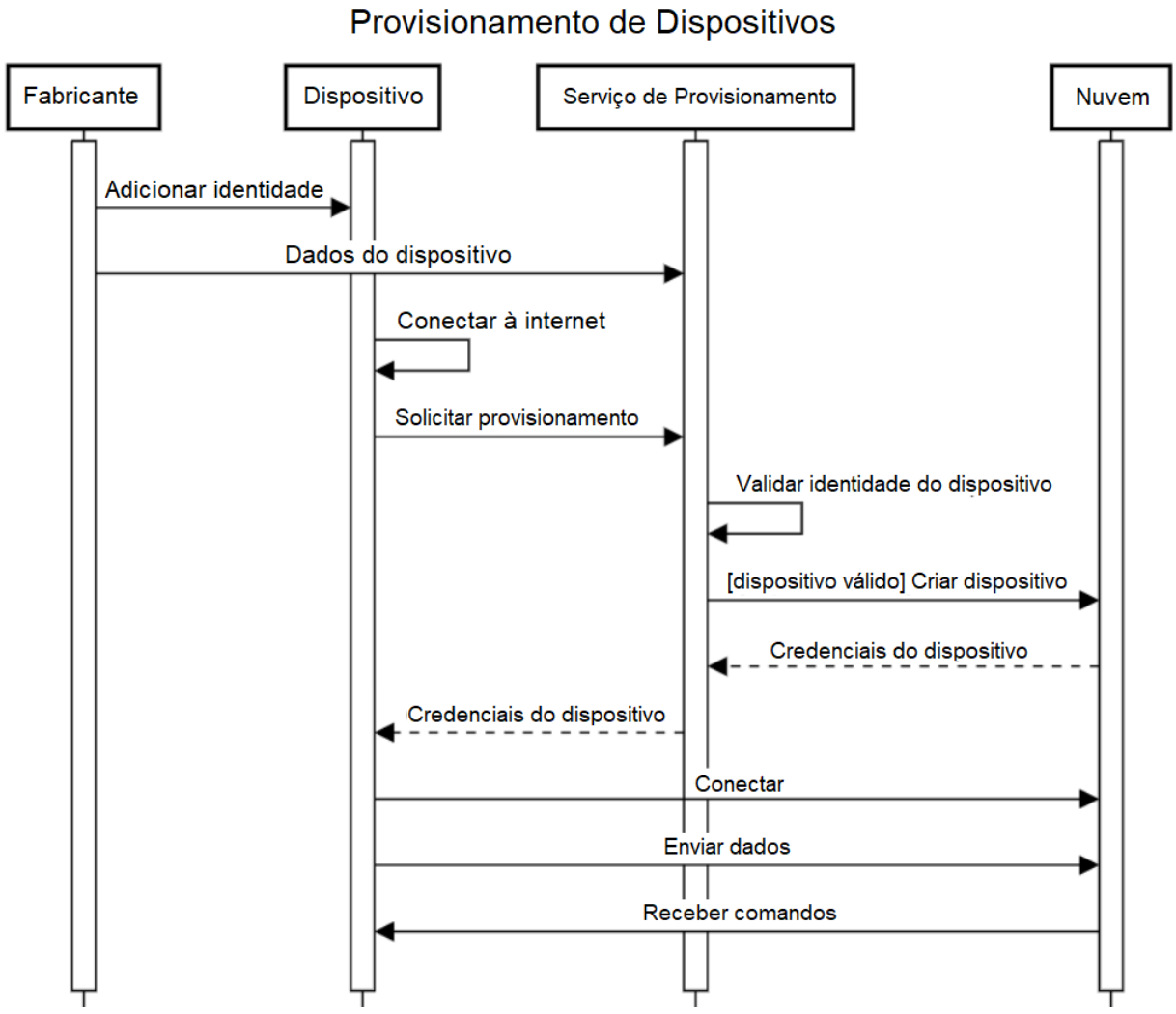

Figura 5.4: Diagrama de sequência para o provisionamento de dispositivos.

1. Fabricante insere dados de identificação no dispositivo para que este saiba a qual serviço se conectar quando estiver online e que identidade ele possui.

2. Fabricante envia os dados da identidade dos dispositivos para o serviço de provisionamento, que os armazena para uso posterior.
3. Quando o dispositivo estiver online, conecta-se ao serviço de provisionamento e envia sua identidade.
4. O serviço de provisionamento valida a identidade e solicita a criação do dispositivo.
5. As credenciais do dispositivo são enviadas de volta para o serviço de provisionamento e depois armazenadas localmente.
6. O dispositivo pode, então, se conectar com o serviço de nuvem para enviar e receber dados de maneira segura.

5.9 Atualizações de dispositivos

OTA (do inglês *Over-the-Air Update*) são atualizações feitas remotamente em que um sistema na nuvem envia uma mensagem a um dispositivo com a versão esperada do software, sendo sua responsabilidade iniciar o download, instalar a atualização, verificar se a versão instalada é a esperada e reportar a sua versão mais recente.

Em IoT, é comum encontrar cenários em que dispositivos são enviados a campo e instalados em localidades remotas, como no alto de torres ou enterrados com tubulações. Dessa forma, não se pode considerar que atualizações de software desses dispositivos sejam efetuadas por meio de acesso físico: o custo dessa operação e seus riscos são impeditivos para esta estratégia.

O diagrama a seguir descreve a sequência de atividades realizadas pelo processo de atualização via OTA.

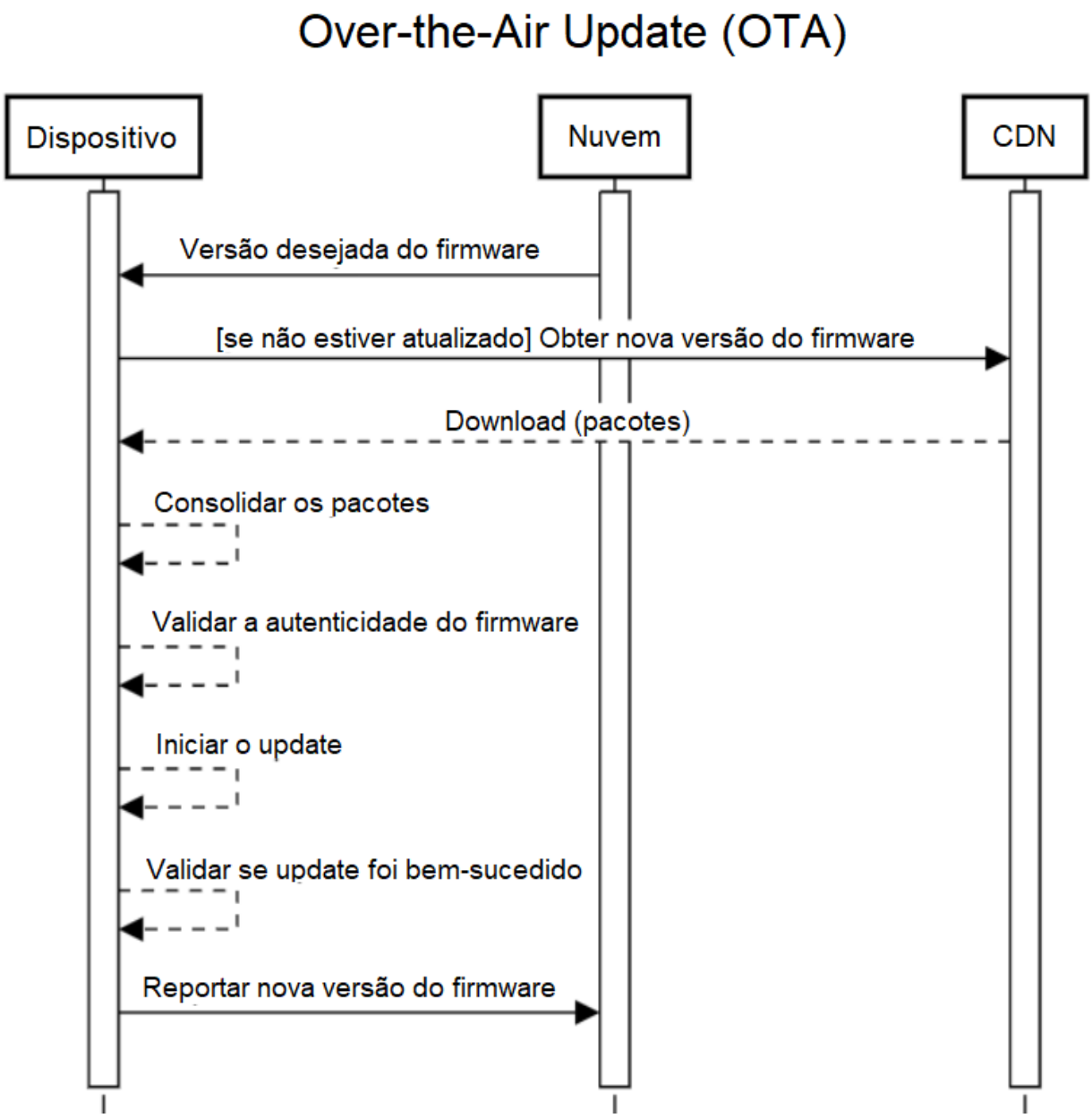

Figura 5.5: Diagrama de sequência para atualizações de dispositivos.

1. O serviço de nuvem envia uma mensagem ao dispositivo com a versão desejada do *firmware*.
2. Caso o dispositivo não esteja executando a versão desejada do *firmware*, ele se conecta ao CDN (em inglês, *Content Delivery Network*) onde os arquivos são armazenados para distribuição e faz o seu download.

3. Após realizar o download, o dispositivo vai recompor o arquivo (se este tiver sido quebrado em partes menores) e validar sua consistência e sua assinatura digital. Todo pacote de atualização deve ser assinado digitalmente para evitar que agentes maliciosos forcem o dispositivo a instalar uma versão não autorizada do *firmware*.
4. O dispositivo então faz a instalação do update e verifica se este processo foi realizado corretamente.
5. Caso a verificação seja bem-sucedida, o dispositivo notifica o serviço na nuvem sobre o update realizado, reportando sua versão atual.

Serviços na nuvem que suportam o processo OTA devem adotar práticas populares de CI/CD (do inglês, *Continuous Integration / Continuous Delivery*) como gerenciamento de imagens de dispositivos, agrupamento lógico de dispositivos (como região, tipo, versão, idioma) e a distribuição de imagens para um grupo alvo de dispositivos. Pode-se citar como exemplo desta prática a distribuição de uma versão do firmware para um grupo de dispositivos em um laboratório de testes antes da atualização dos dispositivos em produção.

5.10 Industrial IoT (IIoT)

Embora a tecnologia que permitiu a explosão da popularidade da internet das coisas seja recente, a automação industrial vem se utilizando de tecnologias muito à frente de outros cenários mostrados neste capítulo, visando aumentar a qualidade dos seus produtos, sua produtividade, a segurança de máquinas e operadores, além de otimizar a produção por meio da automação.

A ilustração a seguir apresenta a pirâmide da automação (PEREIRA, 2022) em fábricas modernas:

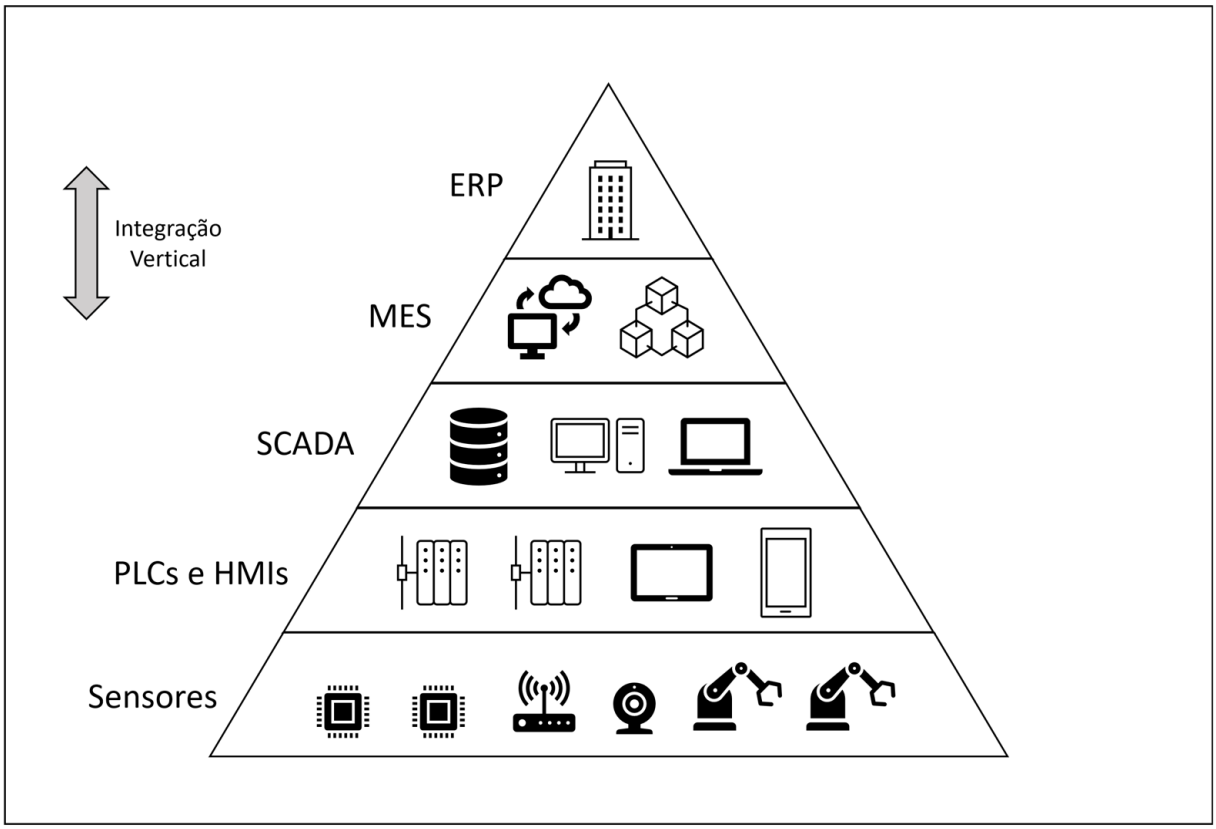

Figura 5.6: Pirâmide de automação industrial.

Sensores e atuadores: sensores podem estar nas máquinas (como sensores de presença de pessoas, temperatura, pressão, viscosidade, vibração, gases, nível de materiais etc.), nas esteiras que levam os produtos entre uma etapa da produção para a próxima (esteiras, elevadores, encanamento, tanques etc.) ou no próprio ambiente onde essas máquinas operam (qualidade de ar, detecção de gases e vazamentos, temperatura e umidade ambiente etc.). Os atuadores, por sua vez, vão controlar essas máquinas: motores, válvulas de operação e de segurança, luzes de indicação de funcionamento, entre outros. Muitas vezes, os sensores e atuadores não estão apenas conectados às máquinas que eles controlam: eles são parte integrante dessas máquinas.

PLCs e HMIs: PLCs (do inglês *Programmable Logic Controllers*, controladores de lógica programável) e HMIs (do inglês *Human Machine Interface*, interface entre humano e máquina) atuam no

nível acima dos sensores e atuadores e efetivamente implementam a lógica que vai controlar o fluxo de produção das máquinas em uma fábrica. Dessa forma, todos os sensores e atuadores mencionados anteriormente estarão conectados a um ou mais PLCs onde toda a lógica de produção está implementada, controlando, por exemplo: a admissão da quantidade correta das matérias primas em um tanque; a mistura dos materiais na velocidade, temperatura e tempo corretos; a monitoração da pressão no tanque, acionando válvulas de descarga de emergência se necessário; e, finalmente, a descarga do produto finalizado para a tubulação que leva o produto final para as máquinas de envase. Todo esse controle passa por uma série de máquinas e dúzias de sensores e atuadores, todos conectados a seus respectivos PLCs. As HMIs fazem parte das máquinas e permitem aos operadores realizar passos da produção como iniciar o processo, monitorar o andamento e acionar o botão de emergência quando necessário.

SCADA (do inglês *Supervisory Control And Data Acquisition*, controle de supervisão e aquisição de dados): como mencionamos no item anterior, o processo de fabricação pode necessitar de várias máquinas e PLCs, mas cada PLC e HMI é responsável apenas por uma parte desse processo. Nesse ponto, os sistemas SCADA entram em ação controlando e definindo o que cada PLC deve fazer ao receber dados de todos os PLCs envolvidos. Os sistemas SCADA geralmente controlam um grande número de PLCs e podem gerenciar toda uma linha de produção de acordo com a necessidade de produção (quantidade e tipo de cada produto que deve ser produzido no momento).

MES (do inglês *Manufacturing Execution Systems*, sistema de execução da fabricação) é um sistema que atua no nível da fábrica como um todo que acompanha e monitora a transformação dos insumos da produção em produtos finais. Além de se comunicar com os sistemas SCADA o MES vai monitorar estoques de matéria prima, sua movimentação pelo processo de produção, tempo de

cada etapa da produção, estoques de produtos finalizados, e todo o pessoal envolvido neste processo.

ERP (do inglês *Enterprise Resource Planning*, sistema de gestão integrado): está no topo de toda a cadeia de automação e auxilia os gestores da empresa a melhorar os processos internos e integrar as atividades de diferentes setores, como vendas, finanças, estoque e recursos humanos. É do ERP que partem as ordens de fabricação de acordo com o que foi vendido a clientes. O ERP integra as informações da empresa numa plataforma única, de forma a facilitar seu compartilhamento e visibilidade enquanto minimiza a duplicidade de informações.

A relação entre a pirâmide da automação com Industrial IoT

O processo de automação descrito representa o estado atual de um imenso número de empresas, mas o calcanhar de Aquiles dessa figura é a necessidade da integração vertical de todas essas soluções: sensores devem estar conectados aos PLCs, que se comunicam com os sistemas SCADA, estes com o MES e, finalmente, com o ERP.

Simples de explicar, mas muito difícil de implementar. A qualidade final da informação e da utilidade de todos esses sistemas depende de todos esses pontos de conexão. Os vendedores de tais soluções podem não ter incentivos para facilitar a conexão de seus produtos com os de concorrentes e facilitar a expansão e atualização de seus sistemas. Essas características acabam prendendo as empresas a um conjunto de provedores de solução, tornando suas atualizações e expansões muito caras. Caso uma empresa opte por adquirir, por exemplo, um PLC de um fabricante e um sistema SCADA de outro, a integração pode ser extremamente complexa, exigindo camadas de tradução de protocolos, o que encarece a solução final e torna a manutenção complexa.

Entra em cena a IIoT, criando a possibilidade de conexão dessas camadas da pirâmide com a nuvem e de utilização de padrões

abertos e interoperáveis (dentre eles, destacam-se o *Industrial Internet Consortium's* (IIC) e o *Industry 4.0*, na Alemanha). Essa tecnologia abre as portas para indústrias que, devido aos altos custos, não haviam conectado toda a sua cadeia de produção, ao mesmo tempo em que cria oportunidade de conexão de dispositivos e sistemas de diferentes fabricantes.

Como mencionado em capítulos anteriores, os dispositivos e sistemas na pirâmide passam a se conectar não uns com os outros (verticalmente) mas, sim, com um sistema centralizado na nuvem (horizontalmente), utilizando, em muitos casos, MQTT para o envio de mensagens de telemetria e recebimento de comandos. Isso torna visível na nuvem a operação de cada PLC, HMI, linha de fabricação e assim por diante, permitindo à organização o acesso instantâneo a dados que antes estavam restritos a sistemas locais.

A conexão dos múltiplos sistemas de fabricação a uma estrutura de IIoT permite a criação de gêmeos digitais (*Digital Twins*) de fábricas que mostram em tempo real o estado de cada um de seus componentes, sendo possível "dar zoom" na fábrica, suas linhas de fabricação, passando por todos os seus PLCs até seus sensores.

A figura a seguir ilustra a relação entre a pirâmide de automação industrial e serviços na nuvem.

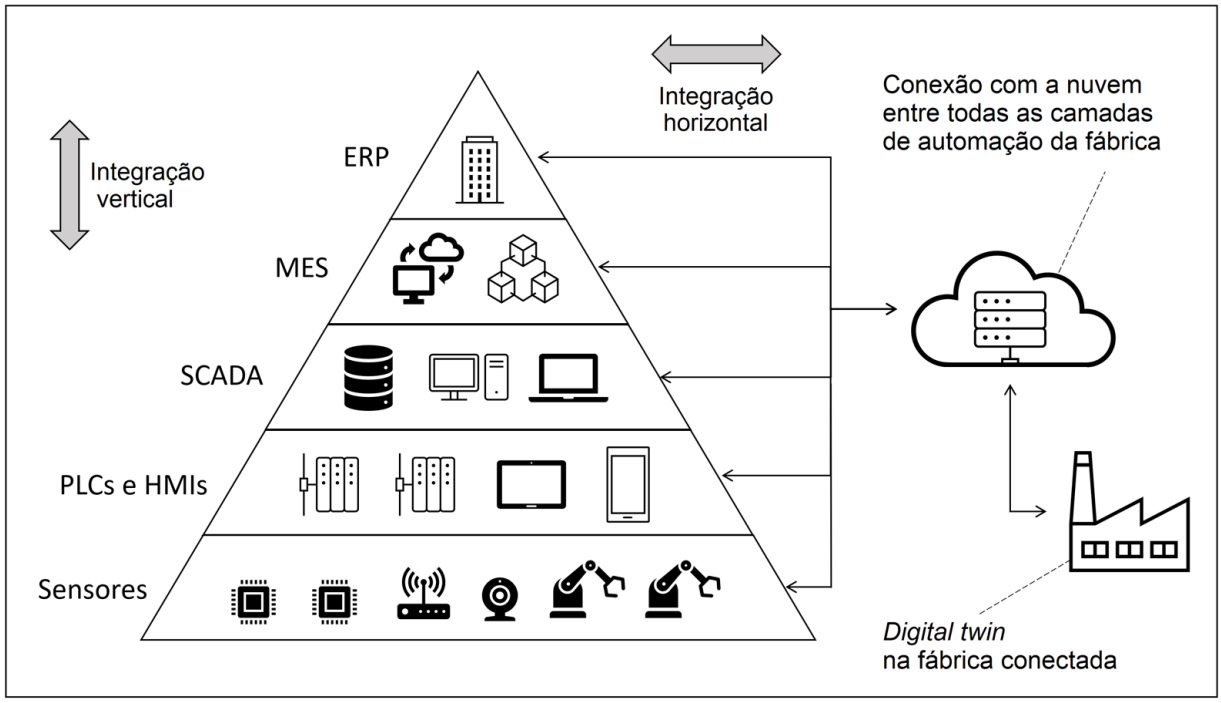

Figura 5.7: Relação entre a pirâmide de automação industrial e serviços na nuvem.

Dispositivos Edge

Dispositivos Edge são encontrados em campo e se comunicam tanto com a nuvem como com outros dispositivos. São chamados de "*Edge*" por estarem na borda/fronteira entre o campo e a nuvem. Comportam-se como servidores e possuem maior poder computacional, além da capacidade de gerenciar outros nós produtores de dados, estendendo suas capacidades ao trazer poder não tradicional para o campo (perto da origem dos dados).

Na maioria dos casos, esses dispositivos não são fisicamente como típicos computadores encontrados em *data centers*. São mais semelhantes a outros dispositivos que também estão em campo, podendo estar em indústrias e fazendas, expostos a poeira e ruídos eletromagnéticos.

O poder computacional dos dispositivos Edge pode ser utilizado de diferentes formas para estender as capacidades em campo, sendo algumas delas:

- Exposição de bancos de dados, sejam eles relacional, não relacional, *time-oriented* ou outros;
- Suporte a soluções de *cache* local de dados;
- Processamento local de imagens baseado em um modelo de aprendizado de máquina (inferência);
- Enriquecimento, agregação e filtro de mensagens;
- Hospedagem de servidor web para visualização local de dados;
- Atuação como *gateway* transparente ou como conversor de protocolos (MQTT para REST, *Bluetooth* para TCP/IP).

Algumas das vantagens em adotar esse tipo de dispositivo são:

- Possibilidade de economizar recursos da nuvem, quando os dados são filtrados e agregados localmente antes de seu envio;
- Funcionalidade independente de conectividade em cenários em que o acesso à internet é limitado. O acesso local aos dados favorece a tomada de decisão em situações críticas ou em manutenções rotineiras em que a comunicação com a nuvem não é possível ou é muito cara;
- Expandir as capacidades no campo ao adicionar recursos de software que antes não seriam possíveis.

O hardware utilizado por dispositivos Edge varia conforme seu cenário de uso. Cenários simples, como conversão de protocolos e *gateway* transparente, podem ser endereçados por microcontroladores. Cenários complexos, como *workloads* que funcionem em *containers* e processamento de imagem, devem considerar a utilização de microprocessadores (sejam esses x86, x64 ou *ARM-based*).

Em geral, provedores de serviços na nuvem têm investido em soluções que suportem cada vez mais a criação de dispositivos Edge, seja na simplificação dos processos de *deployment* e de atualização (utilizando *Over-the-Air Update*), assim como na facilidade em integrar esses dispositivos a seus serviços na nuvem. Tais soluções permitem o provisionamento e execução de *containers*, onde cada um deles pode suportar *workloads* como:

processamento de imagens, execução de um banco de dados, hospedagem de um servidor web ou até mesmo um *container* customizado para o domínio da solução.

A imagem a seguir ilustra um dispositivo Edge atuando como um *gateway*, recebendo mensagens de dispositivos em campo por meio de diferentes protocolos/tecnologias de rede e conectado à internet. Esse dispositivo Edge pode hospedar múltiplos *containers* que adicionam mais recursos à solução em campo.

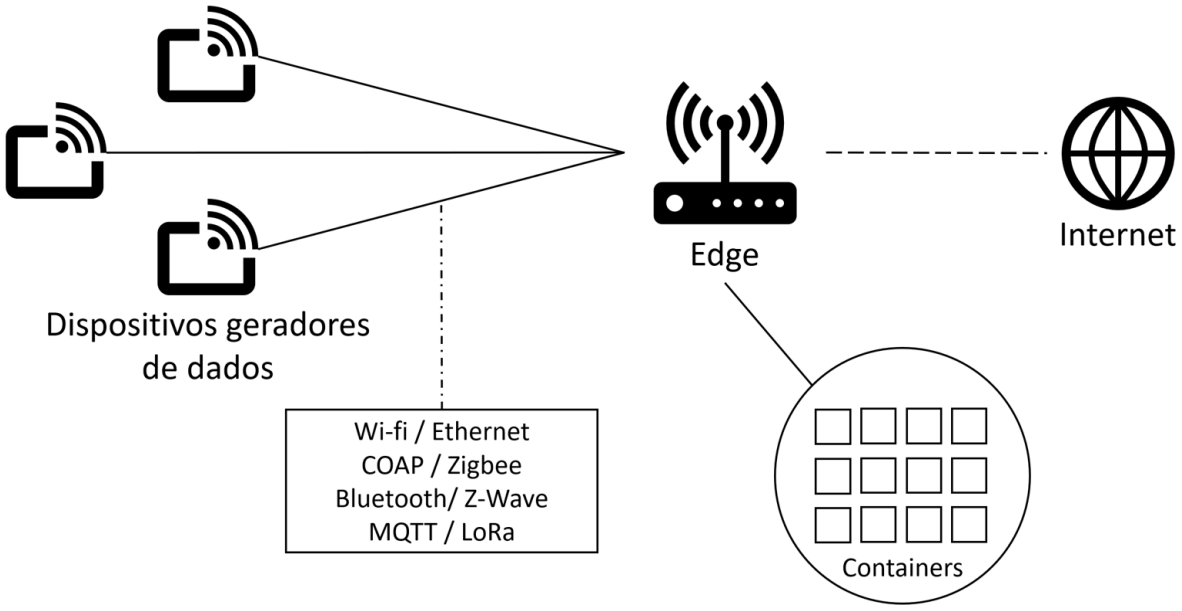

Figura 5.8: Dispositivo Edge hospedando containers e comunicando-se com outros dispositivos em campo e com a internet.

Conclusão

A simples troca de mensagens e comandos (*mensageria*) não é o principal fator que faz a IoT tão poderosa – na verdade, esta é a fundação da IoT. As ferramentas que são construídas com base na troca de mensagens e comandos (tais como OTA, DPS, *Device Twins*) são o motivo pelo qual a IoT evoluiu e tornou-se uma solução viável para tantos novos cenários de negócio.

No próximo capítulo, são apresentados os conceitos e tecnologias descritos anteriormente por meio de cenários de negócio de IoT.

Capítulo 6
Cenários de IoT

Neste capítulo são apresentados cenários comuns de IoT. Cada cenário descreve um processo de negócio e destaca como o uso de IoT o suporta e como seu público-alvo interage com a tecnologia, provendo uma visão horizontal de fim a fim, combinando diversas soluções e aspectos arquiteturais importantes.

Cada cenário de IoT possui vários aspectos arquiteturais que devem ser considerados no desenvolvimento da solução, como: segurança, escalabilidade, eficiência, desempenho e disponibilidade. Cada aspecto arquitetural não representa um cenário por completo, mas apenas uma parte da arquitetura que deve endereçar os vários cenários em uma visão horizontal.

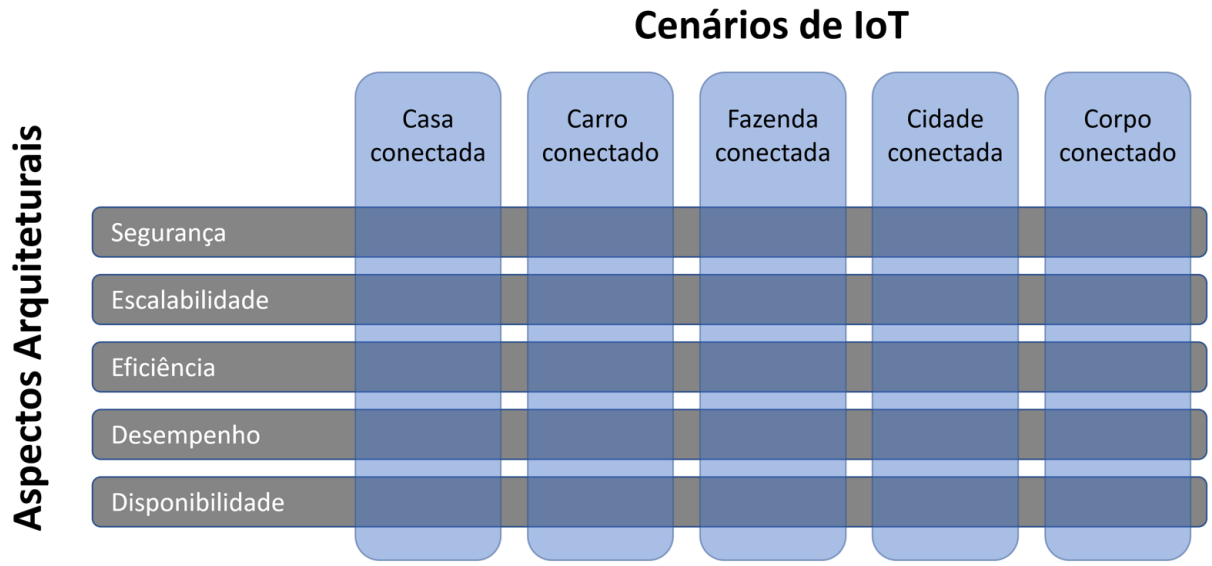

Figura 6.1: Aspectos arquiteturais de cenários de IoT.

Cabe à pessoa arquiteta de IoT identificar, para um cenário específico, quais aspectos arquiteturais devem ser considerados e a sua criticidade. Para cada cenário de IoT, a criticidade de um aspecto arquitetural é variável de acordo com o cenário de negócio.

Por exemplo, é possível que a disponibilidade ou a escalabilidade seja flexível para alguns cenários, mas que, para outros, seja vital.

No contexto deste capítulo, o conceito de ROI (do inglês *Return Over Investment*) é utilizado para descrever de forma resumida os custos e benefícios da implementação dos cenários de IoT descritos.

Neste capítulo serão abordados os seguintes cenários:

1. Casa conectada (casa inteligente, ou *smart home*);
2. Corpo conectado (*smart wearables*);
3. Cidade conectada;
4. Carro conectado;
5. Fazenda conectada;
6. Navio conectado;
7. Fábrica conectada.

6.1 Casa conectada

A casa conectada (ou casa inteligente) possui infraestrutura para permitir a comunicação entre um sistema centralizado e diversos dispositivos, possibilitando o monitoramento e controle remotos. A casa pode, ainda, se autogerenciar seguindo programas preestabelecidos para o controle dos dispositivos.

O cenário de casa conectada pode variar desde o mais básico (tal como uma simples lâmpada inteligente) até o mais complexo (tal como sistemas integrados de automação de temperatura, iluminação, segurança etc).

Público-alvo – todos aqueles que convivem no local: pessoas que buscam comodidade; pais/mães que trabalham fora e querem acompanhar a relação entre filhos e babá; idosos e outras pessoas que necessitam de ajuda emergencial com alguma frequência;

moradores que desejam elevar o nível de proteção de seu patrimônio, entre outros.

Restrições – alguns dos serviços da casa devem continuar funcionando mesmo que a conexão com a internet seja interrompida, como: eletrodomésticos, central de alarme e ar-condicionado. Da mesma forma, a queda de energia também deve ser prevista em alguns casos com dispositivos sendo alimentados por baterias ou inversores.

Redes de comunicação de contingência para cenários críticos podem se utilizar de rede celular como parte da automação, como em sistemas integrados de segurança. Diante de cenários de contingência, alguns serviços da casa podem funcionar com restrições para economizar energia ou banda de internet (por exemplo, reduzir o envio de telemetria e restringir o acesso a funcionalidades), mas ainda prover recursos básicos de suporte à casa.

ROI – do ponto de vista do usuário, o retorno sobre investimento para a casa conectada varia desde a simples comodidade de controlar a sua TV usando um assistente de voz ou de ligar a máquina de café remotamente, até cenários mais críticos como o de assistência a idosos, onde a existência dessa tecnologia pode salvar vidas.

Implementações – ao longo do tempo, mais e mais dispositivos tornaram-se inteligentes. Centrais de multimídia, eletrodomésticos, climatização e até mesmo lâmpadas tornaram-se conectadas para facilitar a vida de seus usuários, sendo controladas por assistentes de voz, que se misturam ao dia a dia do lar e tornam-se uma ferramenta vital para melhorar a qualidade de vida. Ao mesmo tempo, sensores de presença, câmeras e sistemas de segurança, que até pouco tempo atrás eram soluções muito caras para uso residencial, tornaram-se mais acessíveis por conta da IoT.

Redes – as tecnologias de rede mais comuns neste cenário utilizam o padrão *mesh* (via Zigbee e Z-Wave) e wi-fi. Também é comum encontrar soluções que utilizam redes de celular para redundância ou mesmo como conexão principal em localidades remotas (por exemplo, casas de veraneio, chácaras, sítios).

Serviços da nuvem – uma vez que os dispositivos sejam conectados com o serviço de nuvem, este pode prover uma interface para controle e monitoramento remoto de dispositivos. A nuvem também permite a integração entre diferentes plataformas, como: sistema de vigilância da vizinhança, sistema de manutenção do condomínio e integração com sistemas de *e-commerce* para entregas. Complementar a todas as integrações de sistemas, ainda existe o suporte a serviços de emergência por meio de notificações e alertas (e-mail, SMS e chamadas telefônicas).

A figura a seguir apresenta uma potencial integração entre todos os dispositivos, a rede da casa conectada e os serviços na nuvem.

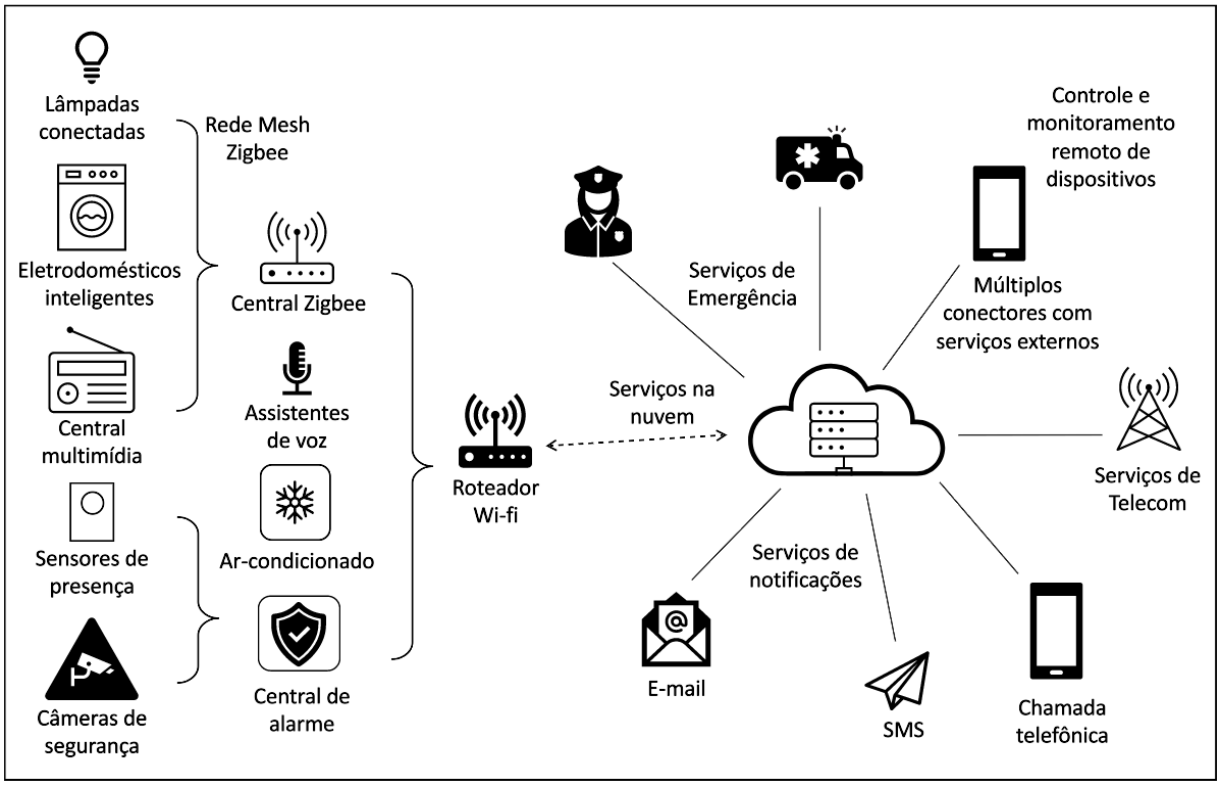

Figura 6.2: Integração entre dispositivos e serviços na nuvem no cenário de casa conectada.

Como arquiteto(a) de IoT, algumas das preocupações relativas a esse cenário são:

- **Dispositivos IoT (sensores e atuadores):** considerar o cenário atual e a possível adição de mais dispositivos no futuro, o uso de padrões abertos de hardware e comunicação, existência de documentação, facilidade de upgrades de hardware, atualizações remotas e facilidade de substituição de unidades defeituosas;
- **Redes de comunicação:** em residências, a existência de uma rede wi-fi disponível é quase sempre algo que o(a) arquiteto(a) de IoT pode considerar, o que não significa que ela seja suficiente. Uma alternativa como redes de celular pode ser necessária para casos em que o wi-fi não esteja disponível ou para cenários em localidades remotas, onde o uso de redes de

celular passa a ser a conexão primária para acesso à internet. A conexão entre os dispositivos pode utilizar-se tanto de wi-fi ou de redes que utilizam o padrão *mesh* (via Zigbee e Z-Wave);

- **Energia elétrica:** no cenário da casa conectada, assim como o wi-fi, a disponibilidade de energia elétrica para os dispositivos é quase sempre garantida, o que facilita o projeto da solução. As exceções são dispositivos que ficam afastados ou em áreas externas e que podem se utilizar de baterias ou de energia solar para funcionar. Dispositivos alimentados por baterias devem ser capazes de notificar o sistema central sobre o estado da bateria para que o usuário possa efetuar a troca antes que o dispositivo pare de funcionar. Dispositivos alimentados por painéis solares podem necessitar de baterias para garantir o funcionamento noturno.

  Deve-se prever alimentação alternativa para todo o sistema (ou parte dele) na ausência de energia elétrica, garantindo a continuidade do funcionamento. A capacidade de operação, nesse caso, depende da criticidade do sistema e, em qualquer caso, é preciso notificar os usuários em circunstâncias adversas, como a perda de energia elétrica.

- **Integrações com sistemas existentes:** a pessoa arquiteta de IoT deve identificar sistemas e soluções preexistentes que possam ser integrados à solução IoT. Essa integração pode ser entre dispositivos e sistemas na nuvem. Cada ponto de integração vai depender de quão aberta é a solução existente, quais protocolos são utilizados e APIs disponíveis. Quando a integração entre dois sistemas não é nativa, a utilização de adaptadores de protocolo se torna necessária.

6.2 Corpo conectado

Corpo conectado é o cenário onde dispositivos e sensores ligados ao nosso corpo geram mensagens de telemetria que são agregadas e enviadas para um serviço na nuvem, permitindo o acompanhamento e monitoração desses dados pelo próprio usuário ou por profissionais de saúde. As possibilidades são muito amplas com a disponibilidade de diferentes sensores.

Público-alvo – o cenário mais comum é o de atletas realizando o acompanhamento de atividades físicas com dispositivos de pulso que medem a frequência de batimentos cardíacos. Existem, também, cenários críticos que podem se beneficiar do monitoramento de sinais vitais, como pacientes de hospitais ou que estão em período pós-operatório em casa; idosos que se utilizam de dispositivos que permitem rastrear sua localização (pacientes com mal de Alzheimer); ou botões de pânico para casos de queda ou de necessidade de solicitar ajuda rapidamente.

Restrições – em cenários cujo foco é proteger a vida do usuário, soluções IoT devem considerar situações extremas e avaliar individualmente o risco de cada parte do sistema, tanto em alimentação do dispositivo (tempo de carga de bateria, velocidade da recarga, alertas e monitoração remota do nível de consumo), quanto em relação à contingência de redes de comunicação (uso de redes de celular) e tolerância a indisponibilidades do serviço na nuvem.

> **NOTA**
>
> No Brasil, soluções que sejam voltadas para ambientes hospitalares ou de monitoração de sinais vitais de pacientes exigem certificações de segurança pela ANVISA (Agência Nacional de Vigilância Sanitária).

ROI – soluções mais simples, como no caso de um atleta amador monitorando sua sessão de exercício, trazem benefícios para a saúde, reforço positivo da atividade e acompanhamento do seu

progresso. Em cenários críticos, como no monitoramento de idosos e pacientes, o correto funcionamento da solução IoT pode ser a diferença entre a vida e a morte de seus usuários.

Implementações – assim como na casa conectada, a quantidade de dispositivos inteligentes para o corpo conectado aumenta a cada dia para os mais diversos fins, como:

- **Relógios inteligentes:** também conhecidos como *smartwatches*, esses dispositivos se comunicam com um telefone celular via Bluetooth e auxiliam em tarefas corriqueiras do dia a dia, como no controle de tarefas, calendários, lembretes, assistentes de voz e leitura de mensagens. Também funcionam como monitores pessoais de pulso, enviando telemetria de batimento cardíaco, quantidade de passos de uma caminhada, níveis de oxigênio no sangue, entre outros dados;

- **Monitores de ondas cerebrais:** esses dispositivos se encaixam na cabeça do usuário e monitoram a atividade cerebral, com diferentes níveis de precisão. São de uso pessoal, comumente utilizados em conjunto com atividades terapêuticas, sessões de meditação, controle de sono, níveis de estresse, entre outros. Assim como os relógios inteligentes, versões mais simples desse dispositivo se utilizam de Bluetooth para comunicação com telefone celular; outras, mais sofisticadas, utilizam conexão via cabo para minimizar o ruído eletromagnético durante a leitura de ondas cerebrais;

- **Roupa inteligente:** à primeira vista, são bermudas, camisetas e jaquetas que parecem ser roupas comuns, mas elas embutem tecnologia que permite o monitoramento de frequência cardíaca, respiração, aceleração, passos, calorias consumidas, temperatura corporal, entre outros. Inicialmente restrito à monitoração de militares e astronautas, essa tecnologia hoje permite que atletas profissionais acompanhem seu rendimento em tempo real;

- **Bombas de insulina:** a possibilidade de conexão com IoT torna estes dispositivos capazes de informar ao usuário seus níveis de açúcar em tempo real, a necessidade de administrar insulina, além de monitorar como as atividades físicas e a alimentação afetam cada indivíduo. Esse dispositivo, diferente dos anteriores, é considerado como equipamento de suporte à vida e, por essa razão, exige rigor em seu desenvolvimento e utilização.
- **Marca-passo conectado:** talvez um dos mais críticos entre os dispositivos conectados, o marca-passo conectado atua como um marca-passo comum: monitora a cadência dos batimentos cardíacos e, ao detectar anomalias, decide pela aplicação de uma alta voltagem para desfibrilar os músculos do coração. A conexão com um telefone celular via Bluetooth permite o monitoramento desses eventos pelo paciente e seu cardiologista, sendo possível monitorar também o estado do próprio marca-passo. Nesse cenário, o consumo de bateria é um aspecto crítico, pois a utilização do Bluetooth não pode acarretar em um significativo consumo da bateria porque a troca de bateria pode exigir um novo procedimento cirúrgico;

Redes – a rede mais utilizada neste cenário é o Bluetooth (ou sua versão de baixo consumo de energia, o BLE), que conecta o dispositivo a um telefone celular que atua como *gateway*.

Serviços da nuvem – assim como os dispositivos da casa conectada, uma vez que os dispositivos do corpo conectado sejam ligados com o serviço de nuvem, este pode prover uma interface para controle e monitoramento remoto de dispositivos, além de suporte a serviços de emergência por meio de notificações e alertas (e-mail, SMS e chamadas telefônicas).

Em alguns casos, relógios inteligentes podem detectar quedas do usuário e, em casos de imobilidade acima de um período, acionar serviços de emergência informando a localização do usuário para o seu auxílio (APPLE, 2021).

A figura a seguir apresenta a possível integração entre dispositivos e nuvem no cenário de corpo conectado.

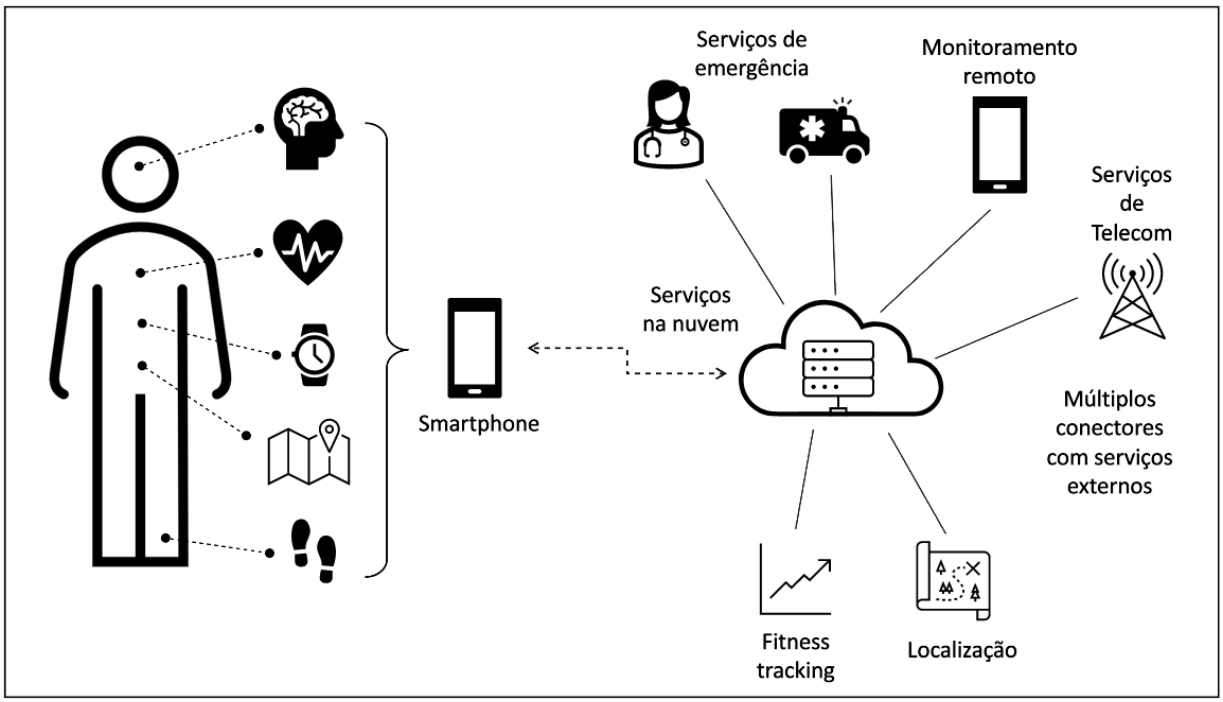

Figura 6.3: Integração entre dispositivos e serviços na nuvem no cenário de corpo conectado.

6.3 Cidade conectada

Cidades conectadas são cidades que utilizam múltiplos dispositivos conectados à internet e que processam, monitoram e analisam dados com o objetivo de melhorar a qualidade de vida de seus cidadãos. É possível dizer que o conceito de cidade conectada não corresponde a uma única solução de IoT, mas à composição de múltiplas soluções que colaboram entre si.

Mais do que conveniente e moderno, as cidades conectadas são uma necessidade: hoje, 54% da população mundial vive em cidades e essas precisam prover serviços de forma economicamente sustentável (THALES, 2022). Dentre as múltiplas possibilidades de

serviços ligados à cidade conectada, temos a melhoria na distribuição de energia, otimização de coleta de lixo, diminuição de congestionamentos, melhoria da qualidade do ar, aumento da segurança, entre outros.

Não apenas grandes metrópoles se favorecem do conceito de cidades conectadas; esse conceito também se aplica a cidades menores, a campus de universidades ou de empresas, clubes e outras entidades distribuídas em múltiplos prédios.

Público-alvo – todos as pessoas vivendo na cidade, mas também visitantes/turistas e todos os órgãos governamentais que atuam na cidade podem se beneficiar das informações providas em tempo real pela rede de sensores de uma cidade conectada.

Restrições – gestores das cidades precisam avaliar cuidadosamente aspectos como de que forma a monitoração da cidade pode ser invasiva na vida de seus cidadãos, o direito do uso de imagem dessas pessoas, o armazenamento dessas informações e a proteção a informações que possam identificar pessoas, endereços etc. Isso não significa, no entanto, que os dados devem ser protegidos contra todo tipo de acesso: na maioria dos casos, apenas os dados que possam identificar pessoas devem ser protegidos, sendo que todas as demais informações podem e devem ser compartilhadas com os cidadãos, via mapas inteligentes, índices de qualidade do ar, letreiros eletrônicos, entre outros. Um exemplo disso é a cidade de Amsterdã, na Holanda, onde os dados capturados são compartilhados com os cidadãos (https://data.amsterdam.nl).

Além da privacidade dos dados coletados, muita atenção deve ser dada especificamente aos atuadores da cidade conectada, ou seja, aos equipamentos espalhados pela cidade que podem ser controlados remotamente como válvulas de água, estações de energia, trens, semáforos, entre outros. Os sistemas que controlam estes equipamentos devem ser segregados de outros sistemas e seu controle deve considerar quesitos como proteção contra

invasões, detecção de invasões e – finalmente – recuperação após os ataques.

ROI – embora os benefícios aos cidadãos e órgãos governamentais sejam incontestáveis, os custos de implementação e manutenção de uma cidade conectada ainda são impeditivos para sua adoção em massa. Cidades piloto como Amsterdã e Nova Iorque são exemplos por serem pioneiras, e seus custos de implantação não foram baixos.

Implementações – a cidade conectada conta com diversas soluções IoT atuando em paralelo e compartilhando dados entre si. Entre elas temos:

- **Controle de rodízio e pedágio urbano:** essa solução é baseada em câmeras que fazem a leitura das placas de veículos para identificar quais não deveriam estar em circulação em determinados dias e horários. Em uma solução similar, as câmeras identificam a quantidade de veículos circulando em uma área específica, variando valores de pedágios urbanos de acordo com a condição atual das vias. Exemplo: em várias cidades nos Estados Unidos, as rodovias possuem faixas exclusivas para veículos com dois ou mais ocupantes que podem circular nessa faixa gratuitamente. Veículos com apenas uma pessoa também podem circular nessas faixas, mas estes precisam pagar uma taxa de pedágio que varia ao longo do dia de acordo com a ocupação da estrada. O valor do pedágio é exibido em letreiros eletrônicos ao longo das rodovias;
- **Monitoramento de desordem pública e eventos:** câmeras que identificam e estimam a quantidade de pessoas em aglomerações, permitindo às autoridades planejar rotas alternativas para o trânsito, estabelecer necessidade de efetivo policial para suporte na área, identificar vandalismo, entre outros;

- **Preço de estacionamento:** sensores de ocupação em estacionamentos auxiliam os motoristas a identificar rapidamente onde as vagas disponíveis se encontram, e também permitem aos operadores ajustar os preços de acordo com a demanda;
- **Transporte público:** a monitoração da frota de transporte público, como a de ônibus, não somente ajuda seus gestores a gerenciar a demanda, mas também auxilia passageiros com a previsão da chegada de ônibus nas diversas paradas;
- **Gestão de energia:** trata da geração, armazenamento, manutenção e distribuição otimizada de energia pela cidade. Um possível exemplo é a adoção de painéis solares distribuídos pela cidade fazendo uso dessa energia para alimentar equipamentos de iluminação pública, sinalização e os próprios dispositivos IoT. Dependendo do tamanho do conjunto desses painéis, a energia gerada pode ainda ser utilizada na recarga de baterias de veículos elétricos;
- **Monitoramento de chuvas e alagamentos:** atuando em conjunto com serviços de previsão do tempo, sensores posicionados em pontos estratégicos da cidade indicam chuvas, alagamentos, nível das águas e podem operar automaticamente bombas de sucção. Esta solução pode ainda notificar serviços de emergência, desviar o trânsito e alertar cidadãos por meio de displays eletrônicos, mensagens de texto e assistentes de voz.

Redes – dispositivos espalhados pela cidade podem se comunicar utilizando diferentes tecnologias:

- **Redes de celular:** para dispositivos instalados em localidades remotas ou de difícil acesso que necessitem de uma maior taxa de transferência de dados. Redes de celular também são utilizadas como uma alternativa de contingência para outras redes;

- **LPWAN:** as redes de baixa potência e longo alcance, como LoRa, são ideais para sensores e outros dispositivos que são alimentados por baterias e utilizam baixa taxa de transferência de dados;
- **Wi-fi:** muito comum em ambientes fechados, mas também pode ser utilizado em ambientes externos quando utilizadas antenas direcionais de wi-fi, que tornam as soluções mais viáveis tanto em termos de custo e transferência de dados, quanto quando comparadas às alternativas de redes anteriores;
- ***Ethernet:*** utilizado quando existe alguma forma de ruído que interfere na utilização de redes sem fio.

Serviços da nuvem – serviços na nuvem proveem uma infinidade de recursos que auxiliam na implementação de cidades conectadas: interfaces de controle e monitoramento dos dispositivos espalhados pela cidade, suporte a serviços de emergência, notificações e alertas, aprendizado de máquina (*machine learning*), *message brokers* e armazenamento dos dados.

Aprendizado de máquina é bastante utilizado em cenários que exigem o processamento de imagens. Uma vez que o modelo esteja treinado, ele pode ser utilizado tanto no dispositivo local que possui poder de processamento para tanto, assim como pode ser executado na nuvem após o upload de uma imagem.

A figura a seguir apresenta possíveis integrações entre dispositivos e a nuvem no cenário de cidade conectada.

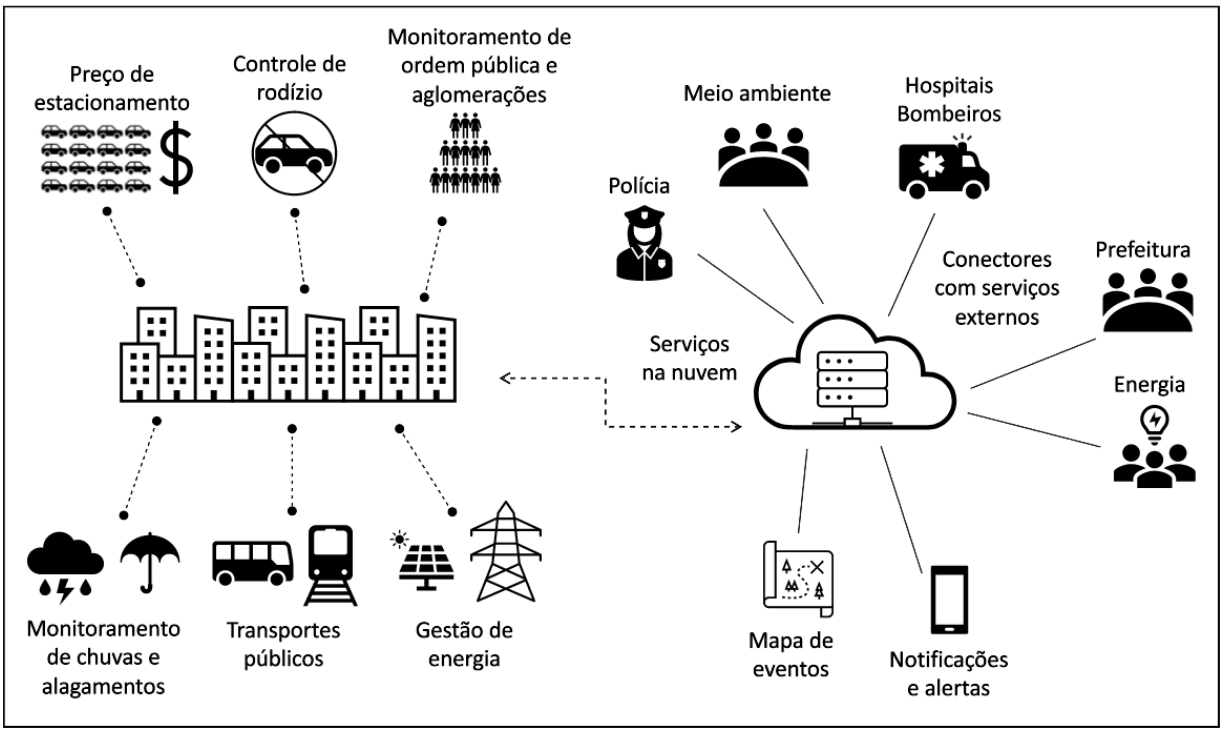

Figura 6.4: Integração entre dispositivos e serviços na nuvem no cenário de cidade conectada.

Como arquiteto(a) de IoT, algumas das preocupações relativas a este cenário são:

- **Dispositivos IoT (sensores e atuadores):** os dispositivos para esse cenário devem suportar ambientes externos e condições climáticas adversas (variações de temperatura, umidade etc.), presença de animais que podem danificar os dispositivos, potencial roubo de equipamentos, entre outros. Dispositivos críticos devem possuir contingência de alimentação e comunicação, além de possuir mecanismos físicos e lógicos de segurança para dificultar sua abertura ou acesso a dados e credenciais.
- **Câmeras:** dispositivos mais complexos como câmeras são necessárias na cidade conectada e, embora eles tenham várias similaridades com os atuadores e sensores, eles geralmente necessitam de maior poder de processamento, maior banda de

comunicação – em alguns casos, elas possuem capacidade de executar modelos de inteligência artificial localmente.

- **Energia elétrica:** sistemas críticos devem prever alimentação alternativa para todo (ou parte) do sistema na ausência de eletricidade, garantindo o seu funcionamento. Assim como citado no cenário da casa conectada, dispositivos alimentados por baterias devem ser capazes de notificar o sistema central sobre estado desta bateria para que seus mantenedores possam efetuar a sua recarga antes que o dispositivo pare de funcionar. Dispositivos alimentados por painéis solares devem utilizar baterias para garantir o funcionamento noturno.
- **Integrações com sistemas existentes:** assim como em outros cenários, a pessoa arquiteta de IoT deve identificar sistemas e soluções preexistentes que possam ser integrados à solução IoT. A cidade conectada possui uma característica diferente dos demais cenários abordados, que é o potencial número de usuários simultâneos. Sistemas integrados a soluções de cidade conectada devem ser escaláveis a fim de atender todos os usuários que precisem utilizar dos recursos providos. Além disso, é importante prover um controle de acesso adequado a cada um destes usuários, sejam estes cidadãos, mantenedores, funcionários de órgãos governamentais ou de empresas terceiras.

6.4 Carro conectado

Veículos têm se tornado cada vez mais inteligentes com o avanço da tecnologia embarcada e na indústria automotiva nas últimas décadas. Esse cenário abrange carros considerados comuns até veículos autônomos que dispensam motoristas. Considerando esses dois extremos, nesta seção é abordada a automação e o valor que a IoT traz para a categoria mais popular de veículos, dado que

veículos autônomos são produzidos com uso massivo de automação baseada em IoT, deixando pouca margem para que outras empresas possam atuar com customizações de hardware.

Ao longo dos anos, os veículos passaram a ter mais tecnologia embarcada ao serem produzidos. Tais tecnologias são utilizadas no gerenciamento do estado do carro e sua deterioração ao longo do tempo. Ainda assim, existem categorias em que os veículos exigem customizações adicionais de hardware para utilizar dos benefícios que a IoT pode prover, como em:

- Veículos agrícolas:
  - Tratores;
  - Colheitadeiras.
- Veículos de transporte público e de passageiros:
  - Táxis;
  - Ônibus urbanos;
  - Ônibus escolares.
- Veículos de emergência:
  - Ambulâncias;
  - Caminhões de bombeiros;
  - Carros de polícia.
- Veículos utilitários:
  - Caminhões;
  - Carro de transporte de valores.

Público-alvo – o público-alvo do carro conectado pode variar dado o produto ou serviço consumido, que leva ao uso direto ou indireto do veículo. Alguns exemplos:

- Comprador de um produto online – acompanhar a geolocalização de um produto comprado online, dada a posição do veículo de entrega;
- Proprietário de um veículo – gerencia a manutenção preventiva, assim como monitora a localização e telemetria do veículo quando utilizados por terceiros;

- Gestor de frota de veículos – gerencia a manutenção preventiva da frota e identifica más práticas dos motoristas;
- Passageiro de transporte público – recebe a estimativa do horário de chegada do transporte público.

Restrições – as restrições se organizam (mas não se limitam) em três categorias:

- **LGPD:** deve-se seguir as normas estabelecidas para proteção de dados pessoais sugeridas pela LGPD (Lei Geral de Proteção de Dados). Todos os cenários de IoT que implicam em rastreamento de pessoas e potencial exposição no vazamento dessas informações exigem cuidados especiais no armazenamento e proteção dos dados que possam identificar pessoas, localizações, endereços etc.;

- **Alterações no veículo:** a implantação das soluções IoT também deve levar em consideração alterações no veículo que podem impactar sua garantia ou alterar suas características, elevando o risco à segurança do veículo ou de seus ocupantes;

- **Segurança de dados:** especial atenção deve ser dada a cenários de segurança, dado o risco inerente de ataques aos sistemas embarcados no veículo. Seus mecanismos de controle devem considerar quesitos como proteção contra invasões, detecção de invasões e recuperação após os ataques.

ROI – ao conectar veículos via IoT, os benefícios adquiridos são: diagnósticos remotos, análise da garantia, análise de consumo e *Over-the-Air Updates*.

O custo da implantação dessas soluções varia de acordo com a complexidade dos dispositivos utilizados: soluções que exigem o envio dos dados em tempo real precisam de comunicação constante (como redes de celular); as demais soluções podem armazenar os dados para o envio quando ao alcance de uma rede wi-fi conhecida.

Implementações

- **Seguradora:** esta solução é baseada em um pequeno dispositivo conectado à porta OBD (do inglês *On-Board Diagnostics*) do veículo, que é normalmente utilizada pelas oficinas para conexão de computadores que fazem a verificação e ajustes nos sistemas computadorizados do motor. Por meio desta conexão, o dispositivo tem acesso a dezenas de tipos de dados referentes ao veículo que são de interesse da seguradora, como: horários em que o veículo é utilizado, quantidade de viagens que o veículo faz por dia, acelerações e frenagens bruscas, velocidade média e máxima do veículo, entre muitas outras. Com esses dados, a seguradora cria o perfil do motorista e oferece descontos aos que considera como sendo bons motoristas. Esse dispositivo normalmente se utiliza da rede celular contratada pelo segurado para o envio de informações, mas o mesmo dispositivo também pode ser utilizado pelo proprietário quando um aplicativo de celular conectado via Bluetooth é capaz de armazenar e mostrar esses dados.

- **Monitoramento de motoristas:** esta solução se utiliza do mesmo dispositivo conectado à porta OBD dos veículos e envia os dados para a nuvem, permitindo o acompanhamento dos hábitos de direção dos motoristas, podendo ainda enviar alertas em casos de colisões.

- **Aquisição de novas funcionalidades:** utilizando *Over-the-Air Update*, o veículo pode ter atualizações em certificados digitais, credenciais, novas versões de *firmware* e de segurança, além de novas funcionalidades e aplicativos que são comercializados após a aquisição do veículo. Um exemplo é o sistema de piloto automático, que pode ser comercializado após a venda por ser uma funcionalidade opcional.

- **Monitoramento geral do veículo:** a monitoração de sinais do veículo como nível e viscosidade do óleo, condição da bateria, estado das velas de ignição, dos filtros e do ar-condicionado, assim como de outras partes do veículo, podem ajudar a

determinar quando a próxima manutenção preventiva deve ocorrer e auxiliar no seu agendamento.

Redes – dada a mobilidade dos veículos, o modelo mais comum de comunicação é via redes de celular, que provê boa capacidade de envio de dados e presença em áreas metropolitanas. Veículos que viajam por regiões onde não exista cobertura da rede celular podem recorrer ao armazenamento local dos dados para o seu envio posterior.

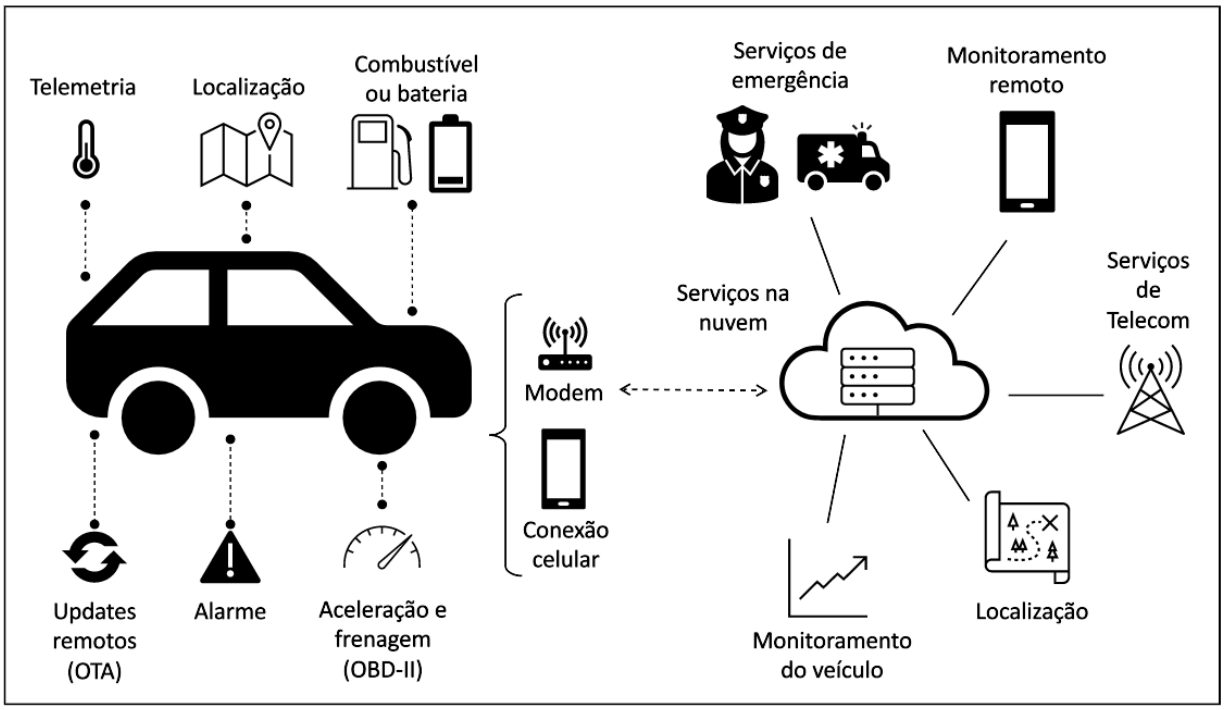

Figura 6.5: Integração entre dispositivos e serviços na nuvem no cenário de carro conectado.

Como arquiteto(a) de IoT, algumas das preocupações relativas a esse cenário são:

- **Dispositivos IoT (sensores e atuadores):** atualmente, os veículos possuem uma porta de comunicação OBD-II, onde todas informações de seus sensores estão disponíveis. Soluções de mercado podem utilizar esta porta para leitura dos

valores desses sensores, assim como para atualização de *firmware*.

- **Segurança:** o maior risco de veículos conectados é que agentes maliciosos tomem o controle sobre o sistema embarcado, colocando em risco a vida do motorista e passageiros.

6.5 Fazenda conectada

Dado o aumento da população mundial, uma preocupação constante é a geração de alimentos de forma eficiente, com a otimização do consumo de recursos como água e espaço, minimizando desperdício e uso de pesticidas e agrotóxicos. Nesse campo de produção de alimentos, além do modelo tradicional de agricultura e pecuária, um conceito recente é o da fazenda vertical (TECNOLOGIA NO CAMPO, 2020), que se utiliza de técnicas de cultivo hidropônico e luzes artificiais para acelerar o crescimento das plantas e que contam, desde sua concepção inicial, com a monitoração e controles automatizados.

Público-alvo – fazendeiros, cooperativas agrícolas, empresas e funcionários que se dedicam na produção de produtos agropecuários.

Restrições – devem-se tomar precauções contra possíveis ataques aos sistemas da fazenda que gerenciam a sua rotina e que podem colocar em risco a vida de funcionários e animais, equipamentos, máquinas e a qualidade de sua produção.

ROI – o custo de implantação de uma solução IoT em grandes áreas como fazendas é alto, mas, uma vez que a infraestrutura básica esteja implementada (por exemplo, antenas receptoras que permitem que transmissores sejam utilizados no campo), adicionar mais unidades tem um custo menor.

O investimento em tecnologia é um caminho inevitável dada a concorrência e a necessidade de aumento na produtividade.

Implementações

- **Monitoramento e controle de plantações:** sensores de temperatura e umidade do solo indicam a necessidade de água e o sistema da fazenda controla a irrigação. Idealmente, esse sistema deve ser conectado com previsões meteorológicas para otimizar recursos; caso exista a possibilidade de chuva, a irrigação pode ser evitada. Outros sensores podem indicar a presença de condições adversas de tempo como geadas, excesso de chuva, excesso de pesticidas, identificação de pragas, entre outros.
- **Monitoramento e controle de silos de grãos:** silos de grãos estocam o resultado de colheitas e seu armazenamento envolve controles rígidos das condições dentro do silo como:
    - Umidade e temperatura – o excesso de umidade pode levar ao apodrecimento dos grãos e mudanças drásticas de temperatura afetam sua qualidade. Desta forma, potentes ventiladores na base do silo são utilizados para controlar essas duas variáveis;
    - Presença de gases específicos – indicam a existência de pragas nos grãos;
    - Acúmulo de poeira – podem acarretar em explosões no silo (PECOSQUI, 2019).
- **Monitoramento e controle de estufas:** as estufas (ou sistema de cultivo protegido) são a base do cultivo orgânico e hidropônico de hortaliças, tomates, uvas e muitos outros vegetais. Esses ambientes demandam controles rígidos de temperatura, umidade, nutrientes presentes no solo, iluminação solar e reciclagem do ar, permitindo o cultivo de certas espécies no decorrer de todo o ano, independente da estação. Para criar as condições ideais de crescimento de cada tipo de vegetal, são necessários sensores que monitoram todas essas

condições e atuadores que vão controlar as condições desse ambiente.

- **Monitoramento e controle de rebanhos:** cada animal possui uma coleira eletrônica com sensores que o identificam no rebanho, monitorando sua localização, temperatura, peso e padrão de alimentação. Estes sensores permitem ainda o controle de porteiras que dirigem os animais para diferentes setores da fazenda, facilitando, por exemplo, o trabalho de um veterinário.

Redes – em uma fazenda conectada diferentes tecnologias de redes serão utilizadas, sempre dependendo da localização do sensor, quantidade de dados trafegados, distância da central da fazenda e movimentação de objetos. Alguns casos de uso são:

- **Controle de rebanho:** as coleiras eletrônicas são equipadas com tecnologias de comunicação de curta distância, pois são econômicas com relação ao uso de bateria e suficientes para a troca de informação com receptores estrategicamente espalhados pela fazenda. Nesse caso, podem ser utilizados BLE, Z-Wave, Zigbee ou RFID;

- **Controle de armazenamento e de plantações:** dada a ausência de energia elétrica, baixo tráfego de dados e potencial longa distância entre as plantações, estufas e silos e a central da fazenda, tecnologias de comunicação LPWAN devem ser utilizadas;

- **Central da fazenda:** além das tecnologias de rede mais populares de acesso à internet como *ethernet* e wi-fi, ainda é necessário o suporte para outras tecnologias de IoT, como a instalação dos *gateways* de LoRa, BLE, Z-Wave, Zigbee e demais tecnologias.

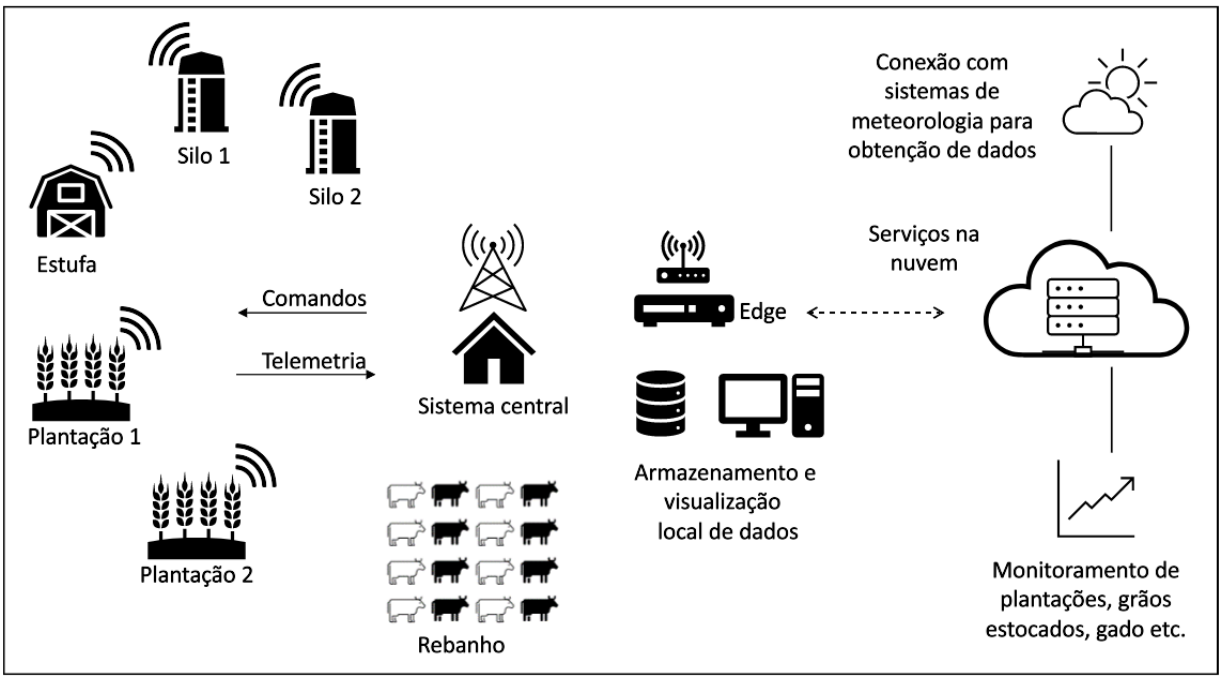

Figura 6.6: Integração entre dispositivos e serviços na nuvem no cenário de fazenda conectada.

Como arquiteto(a) de IoT, algumas das preocupações relativas a esse cenário são:

- **Dispositivos IoT:** geralmente estão em locais afastados, expostos a condições climáticas adversas, animais e insetos. Considere soluções que forneçam vedação dos dispositivos para suportar as condições, de acordo com as especificações dos componentes e microcontroladores.
- **Disponibilidade de energia elétrica:** como os dispositivos estão instalados em locais remotos, onde não há disponibilidade de energia, é preciso adotar soluções alternativas. Dispositivos que consomem pouca energia ou que têm baixo tráfego de mensagens por dia podem utilizar baterias; em contrapartida, dispositivos que consomem mais energia devem utilizar painéis solares e baterias recarregáveis.
- **Funcionamento offline:** o sistema central da fazenda deve ser capaz de funcionar sem conexão com a nuvem, seja por uma

questão de economia de tráfego de dados ou de custo, seja em razão de possíveis interrupções no serviço de internet em zonas rurais. Essa solução deve ser capaz de receber dados dos sensores e armazená-los localmente para que opere sem interrupções. A conexão com a nuvem, quando restabelecida, permite o sincronismo dos dados entre a nuvem e a solução local.

6.6 Navio conectado

Navios cargueiros e de transporte de *commodities* (como grãos e petróleo) são cruciais na movimentação de bens no comércio global. Estima-se que 80% do transporte de bens é feito por vias marítimas em um valor de 4 trilhões de dólares anuais. Sua importância para o comércio mundial é tão grande quanto seu potencial impacto ambiental em termos de poluição, decorrentes de vazamentos, naufrágios e outros possíveis problemas. Embora essa indústria seja alvo de automação há várias décadas, a inclusão de IoT tem permitido maior inovação e controle das embarcações.

As empresas que adotaram essas soluções disruptivas alcançaram um diferencial competitivo ao reduzir o montante de trabalho manual, aperfeiçoar o controle dos navios e de sua carga, prover informações em tempo real a seus clientes e minimizar incidentes.

Público-alvo – funcionários da empresa de transporte marítimo que proveem suporte em terra, a tripulação que trabalha na embarcação e clientes que contrataram o serviço.

Restrições – adotar mecanismos de proteção contra ataques maliciosos às embarcações que possam acarretar na manipulação de instrumentos de navegação que levem ao risco da integridade física da tripulação, assim como da carga transportada. Adicionalmente, devem-se adotar medidas de proteção da privacidade e de segurança da informação da carga transportada.

ROI – o custo de implementação de uma solução IoT em embarcações marítimas é alto devido ao uso de sistemas globais de comunicação, tais como os sistemas providos por satélite. Mas o retorno sobre o investimento tende a ser rápido, devido aos benefícios e diferenciais competitivos em adotar tais soluções.

Implementações

- **Localização de contêineres no navio:** por meio de *BLE beacons*, é possível identificar a localização dos contêineres dentro do navio, determinar se este foi entregue ou até mesmo se foi comprometido por situações adversas como alagamento, temperatura excessiva etc.
- **Monitoramento dos contêineres:** cada contêiner pode ser monitorado por meio de diversos sensores (temperatura, umidade, altitude, iluminação, presença de gases etc.). Isso permite determinar, por exemplo, se contêineres refrigerados estão funcionando normalmente – alimentos, bebidas e remédios podem ser afetados por variações de temperatura ou se o mecanismo de refrigeração parar de funcionar. Sensores de vibração ajudam a determinar se o contêiner sofreu um choque ou queda e o momento em que isso aconteceu.
- **Monitoramento dos tanques:** navios petroleiros e de transporte de grãos não se utilizam de contêineres, mas sim de imensos tanques onde a produção é armazenada. Nesse cenário, dezenas de sensores monitoram as condições desses tanques, medindo sua capacidade, presença de vazamentos, infiltração e/ou contaminação da carga, temperatura, umidade, presença de gases específicos e muitos outros dados que permitem à empresa e a seus clientes acompanharem (às vezes em tempo real) a condição de seus produtos.
- **Monitoramento e controle do navio:** ao longo dos anos, os instrumentos de navegação e monitoramento utilizados por embarcações foram atualizados e, com o advento da IoT, novas possibilidades foram adicionadas, facilitando a integração destas embarcações em alto mar com suas equipes em terra. A adição de IoT também permite que embarcações se comuniquem com sistemas externos que proveem monitoramento em tempo real sobre a rota da embarcação, as

condições gerais dos navios, as previsões do tempo e fatores externos.

Redes – a tecnologia de comunicação para troca de dados mais utilizada em embarcações é a comunicação via satélite, que, embora tenha cobertura global, tem custo elevado. Por esse motivo, apenas a comunicação essencial para o funcionamento e a segurança do navio são transmitidas via satélite. As demais informações podem ser armazenadas localmente para transferência em um dos portos onde o navio atraca. Adicionalmente, redes *ethernet* e wi-fi são utilizadas para o tráfego de dados até a central da embarcação.

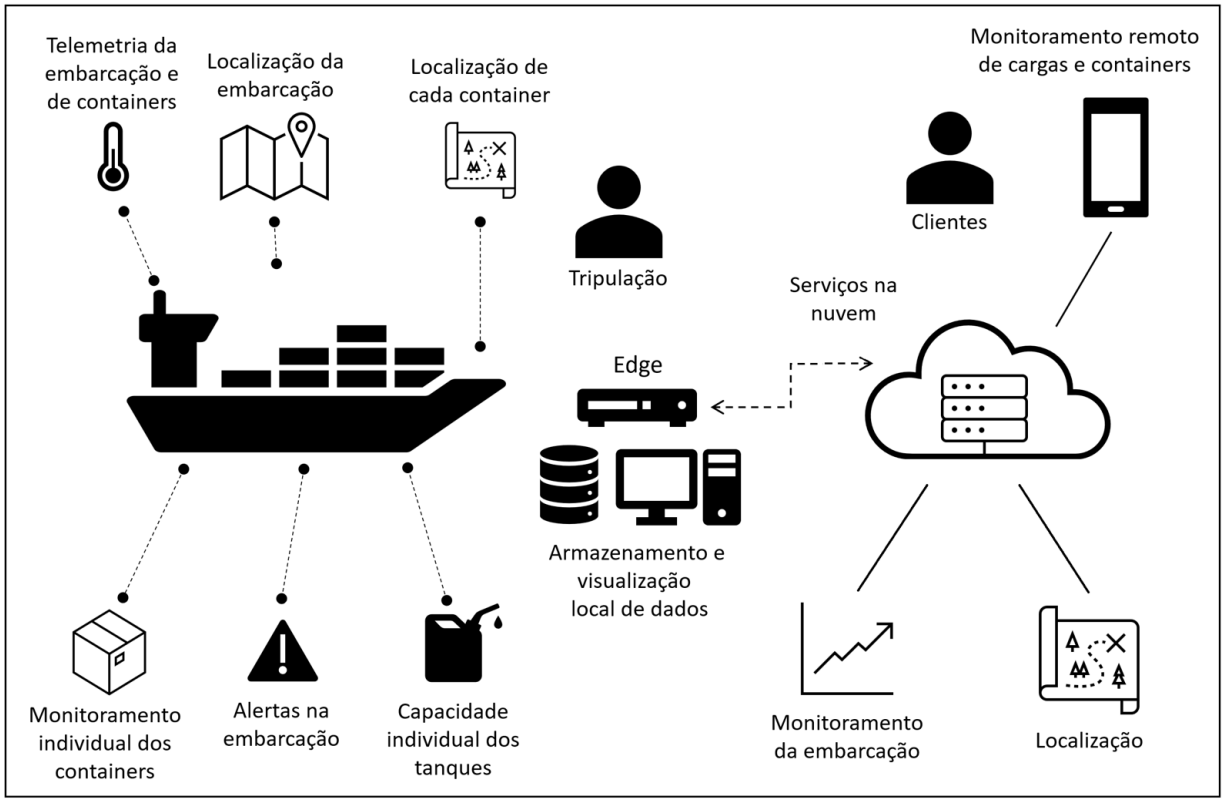

Figura 6.7: Integração entre dispositivos e serviços na nuvem no cenário de navio conectado.

Como arquiteto(a) de IoT, algumas das preocupações relativas a este cenário são:

- **Dispositivos IoT:** geralmente estão em locais afastados no navio, expostos a condições climáticas adversas e produtos que podem ser corrosivos ou tóxicos. Esses dispositivos devem ser capazes de suportar essas condições e manter seu funcionamento normal. Considere especificações dos componentes e microcontroladores como margem de segurança, vedação contra água e produtos químicos que podem afetar o dispositivo.
- **Eletricidade:** a bordo do navio, existe disponibilidade de eletricidade gerada na embarcação, embora cenários de contingência devam ser considerados. Uma vez que a operação normal, assim como a contingência, são providas pelos próprios sistemas da embarcação, o(a) arquiteto(a) de IoT deve considerar essas alternativas.

6.7 Fábrica conectada

Como já mencionado em *Industrial IoT*, a modernização de ambientes industriais não é recente e suas máquinas contam com intensa automação e controle, permitindo o aumento de qualidade e produtividade. A IoT permite expandir os limites da automação enquanto otimiza a eficiência e os custos de produção.

Público-alvo – funcionários que atuam no processo de produção, executivos responsáveis pelas fábricas conectadas e os clientes que indiretamente usufruem do aumento de qualidade dos produtos e de potencial redução no seu custo final.

Restrições – ambientes industriais podem apresentar diversas áreas de riscos ambientais e aos profissionais que nela trabalham. A adição de dispositivos IoT em ambientes industriais requer avaliação desses riscos e sua mitigação apropriada.

Em diversos setores industriais que produzem alimentos e equipamentos hospitalares, ou com alto risco à vida (como metalurgia), rígidas normas governamentais que regem esses setores devem ser seguidas, o que se aplica a quaisquer componentes adicionais utilizados no processo de produção, como sensores e atuadores IoT. Esses equipamentos podem ainda requerer certificados e adequação às normas de segurança antes de serem utilizados em ambientes industriais.

ROI – o custo de implementação de uma solução IoT em uma fábrica tende a ser especialmente alto por requerer adequação às normas de segurança e pelo uso de dispositivos em ambientes hostis, com a presença de gases, temperaturas extremas, poeira, vibração e ruído eletromagnético causado por motores. O retorno do investimento se dá por meio dos ganhos em produtividade e qualidade que tais soluções trazem. A utilização de equipamentos apropriados em ambientes industriais faz com que estes tenham sua duração prolongada, mesmo considerando o ambiente em que atuam.

Implementações

- **Monitoração da vibração de motores:** uma das características mais comuns de motores industriais é sua vibração quando em funcionamento. Sensores posicionados junto a esses motores são capazes de detectar variações ao longo do tempo, indicar problemas e permitir a manutenção preventiva, por exemplo, se um rolamento apresentar sinais de desgaste, o que aumenta a vibração no motor.
- **Conexão de equipamentos legados (dispositivos *brown field*):** uma técnica muito comum na modernização de fábricas é a conexão de maquinário antigo com a IoT por meio de *gateways*. Tais gateways atuam como tradutores de protocolo como CANbus e Modbus para outros protocolos, como MQTT.

- **Controle de qualidade:** dispositivos IoT baseados em câmeras estão sendo amplamente utilizados na detecção de anomalias de produtos em movimento nas esteiras das fábricas. Tais dispositivos têm capacidade de executar inferência localmente sem a necessidade de enviar as imagens para a nuvem para sua avaliação. Modelos de aprendizado de máquina são previamente treinados e enviados a estes dispositivos, que requerem, ainda, câmeras com qualidade condizente aos produtos que se necessita avaliar.

Redes – fábricas tendem a ser ambientes hostis para as tecnologias de comunicação sem fio como wi-fi em virtude da interferência gerada pelos campos eletromagnéticos de motores e da fiação que atravessa a fábrica. Por esse motivo, é muito comum a utilização de redes cabeadas como *ethernet* ou mesmo fibras óticas, que são menos suscetíveis a tais interferências.

Enquanto o cabeamento soluciona grande parte dos dispositivos IoT em uma fábrica conectada, ele ainda não elimina a necessidade de suporte a dispositivos que requerem mobilidade como robôs, empilhadeiras, dispositivos de interface humana (HMI) portáteis etc. Para esses casos, é necessário o uso de redes sem fio como o wi-fi e que tenham capacidade de suportar a operação em ambientes como o da fábrica; tais equipamentos são mais robustos e operam com maior potência, diferentemente dos de categoria doméstica. Deve-se ainda prever a presença de múltiplos pontos de acesso para garantir a cobertura de sinal em todas as áreas onde se fará uso de equipamentos sem fio.

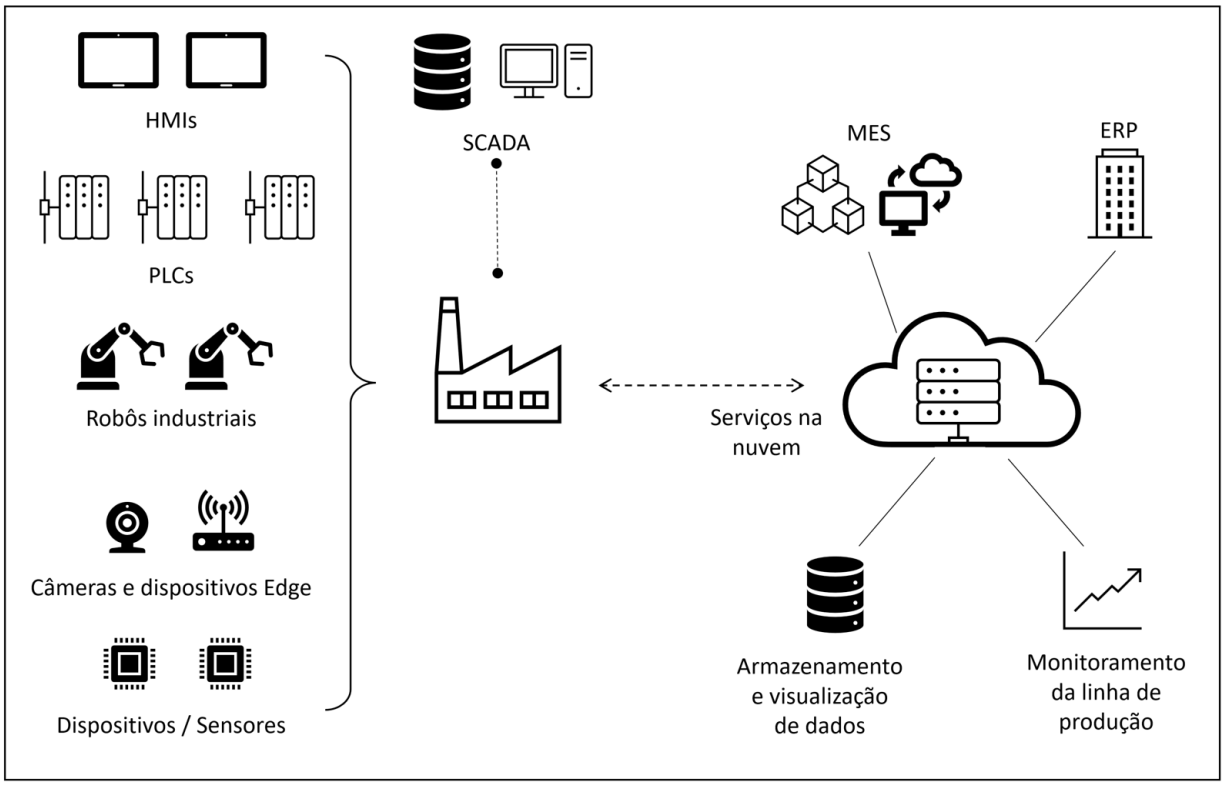

Figura 6.8: Integração entre dispositivos e serviços na nuvem no cenário de fábrica conectada.

Como arquiteto(a) de IoT, algumas das preocupações relativas a este cenário são:

- **Dispositivos IoT:** embora as condições dentro de fábricas variem imensamente de acordo com o produto fabricado e suas matérias-primas, esses dispositivos IoT estão frequentemente expostos a condições adversas e produtos que podem ser corrosivos ou tóxicos, poeira, vibração, temperatura etc. Esses dispositivos devem ser capazes de suportar essas condições e manter seu funcionamento normal. Dispositivos com padrão industrial são muito mais robustos que similares para uso comercial e devem ser preferidos neste cenário.
- **Eletricidade:** tecnicamente falando, alimentar dispositivos elétricos numa fábrica não deveria apresentar um desafio, dado que praticamente todas as máquinas são elétricas. No entanto,

o próprio fato de diversas máquinas compartilharem a mesma rede elétrica representa um desafio, pois quando motores elétricos estão trabalhando, eles naturalmente geram interferências eletromagnéticas que podem “poluir” a rede elétrica em que estão conectados. Neste caso, dispositivos baseados em microcontroladores podem ser afetados por tais interferências, chegando até mesmo a impedir seu funcionamento. A utilização de filtros na rede elétrica ou de redes elétricas segregadas são alternativas para os casos mais críticos.

Conclusão

Neste capítulo, foram apresentados cenários de uso de IoT e características importantes que devem ser consideradas por pessoas arquitetas ao planejar tais soluções.

No próximo capítulo, são apresentados desafios arquiteturais relevantes no planejamento de qualquer solução de software, mas com a adicional perspectiva exigida por soluções IoT.

Capítulo 7
Desafios arquiteturais de uma solução IoT

Soluções IoT apresentam desafios arquiteturais peculiares se comparadas a outros tipos de soluções que são hospedadas em ambientes controlados, com rede cabeada, temperatura sob controle, barreiras de segurança lógicas e físicas e equipes de manutenção próximas fisicamente.

A título de exemplo, considere uma solução IoT cujos sensores de um dispositivo coletam dados de um ambiente, os envia para a nuvem, que, por sua vez, baseada em uma lógica de negócio, envia de volta para o dispositivo comandos que serão executados por seus atuadores. Por um lado, contamos com o poder computacional altíssimo dos provedores de serviços de nuvem, que possuem alta disponibilidade, recuperação de desastres, elasticidade, segurança etc.; por outro lado, temos dispositivos extremamente frágeis e não seguros de uma solução, estando suscetíveis a todo tipo de ameaça ou intempérie, tanto lógicas como físicas.

O objetivo deste capítulo não é esgotar as soluções para todos os desafios arquiteturais de uma solução IoT, mas apresentar atributos de qualidade importantes em sua arquitetura.

De acordo com Bass *et al.* (2021), muitos atributos determinam a qualidade da arquitetura de um sistema. Um deles é a "funcionalidade" que descreve a capacidade do sistema de fazer o trabalho para o qual foi projetado.

Embora a funcionalidade e outros atributos de qualidade estejam intimamente relacionados, normalmente a funcionalidade é priorizada no processo de desenvolvimento. Essa preferência, entretanto, é míope: apesar de atenderem às funcionalidades esperadas, sistemas são reprojetados por serem difíceis de manter,

de portar para outras plataformas, não serem escaláveis, por terem problemas de desempenho ou de comprometimento da segurança.

Um atributo de qualidade é uma propriedade mensurável ou testável de um sistema, que indica o quanto o sistema satisfaz as necessidades de seus *stakeholders*.

Os atributos de qualidade dificilmente podem ser alcançados isoladamente. A realização de qualquer um poderá afetar na realização de outros, tanto positivamente como negativamente. Essa relação entre os atributos é conhecida como *trade-offs*. Por exemplo, a implementação de várias camadas de segurança em um sistema pode impactar negativamente o seu desempenho. Recomenda-se que a escolha dos *trade-offs* seja definida pela necessidade de negócio, considerando tanto os requisitos funcionais como os não funcionais.

Atributos de qualidade em uma solução IoT

Em relação aos atributos de qualidade de uma solução IoT, o(a) arquiteto(a) deverá:

- Identificar quais são os atributos de qualidade necessários para atender aos requisitos de negócio da solução;
- Analisar os possíveis *trade-offs* envolvidos nas decisões tomadas;
- Documentar decisões e as razões para cada atributo;
- Determinar como os componentes da solução vão reagir a situações em que os atributos de qualidade não possam ser mais alcançados e detectar, agir e se recuperar desses eventos.

Tanto quanto possível, essas aplicações devem se basear em circuitos com *feedback* onde cada ação (especialmente as mais críticas) pode ser confirmada e ter seu tempo de execução medido e reportado.

Alguns dos atributos de qualidade mais importantes de uma solução de IoT (BASS *et al.*, 2021) são apresentados neste capítulo, entre eles: funcionalidade, desempenho, disponibilidade, usabilidade, segurança, escalabilidade, manutenibilidade e eficiência no consumo de energia.

7.1 Funcionalidade

Funcionalidade é a capacidade do produto de software de fornecer funções que atendam às necessidades explícitas e implícitas quando usado sob condições especificadas (ISO 25010, 2011). Embora a funcionalidade e demais atributos de qualidade estejam intimamente relacionados, a funcionalidade geralmente tem maior prioridade no processo de desenvolvimento.

Segundo a norma ISO 25010 (2011), o atributo de funcionalidade possui uma característica que representa o nível em que as funções de um sistema atendem às necessidades de negócio e que, por sua vez, é dividida em três subcaracterísticas:

- Completude funcional – o quanto o conjunto de funções atende a todas as tarefas especificadas e objetivos do usuário;
- Correção funcional – o quanto as funções proveem os resultados corretos com o grau de precisão necessário;
- Adequação funcional – grau de adequação em que as funções permitem a realização de tarefas e objetivos.

Requisitos funcionais descrevem o que o sistema deve fazer. De forma análoga, requisitos não funcionais descrevem atributos de qualidade que não são representados através de funções e que possuem grande influência na percepção da qualidade do sistema pelo usuário final. A título de exemplo:

- Requisito funcional – um dispositivo deve fazer a cada hora a leitura da temperatura de uma sala em graus Celsius, com

precisão mínima de 95%.
- Requisito não funcional – apesar de a leitura da temperatura acontecer a cada hora, falhas de conexão impedem que os dados sejam enviados para a nuvem, tornando os usuários do sistema insatisfeitos.

Requisitos funcionais podem abranger requisitos não funcionais implicitamente. Dessa forma, pode-se afirmar que requisitos funcionais e não funcionais estão interligados, visto que a implementação de uma funcionalidade pode envolver requisitos não funcionais (ANDRADE, 2015). Por exemplo, uma câmera inteligente deve conter mecanismos que impeçam a interceptação de seu *streaming* de vídeo por usuários maliciosos.

7.2 Desempenho

O atributo de qualidade de desempenho está relacionado à capacidade do sistema de atender aos requisitos de tempo para executar uma ou mais tarefas (BASS *et al.*, 2021). Em outras palavras, é a capacidade de executar uma função dentro de um intervalo de tempo determinado.

Pode-se medir o desempenho de duas maneiras:

- Latência: o tempo consumido para responder a um evento (tempo de resposta);
- Capacidade (do inglês *throughput*): representa o número de eventos que podem ser processados em um intervalo de tempo.

Latência e capacidade são duas variáveis presentes em toda solução e que, quando otimizadas, favorecem o seu desempenho como um todo. Da mesma forma, toda parte da aplicação pode influenciar positiva ou negativamente o seu desempenho. Por exemplo, a escolha incorreta de um adaptador de rede para um

dispositivo IoT pode impactar o desempenho da solução, independente de quão poderosa seja a infraestrutura na nuvem.

É de responsabilidade do arquiteto determinar qual a criticidade e o desempenho desejados para cada parte da solução, mesmo que não sejam expressos ou explicitamente definidos pelos usuários.

Exemplo de uso – Válvula industrial de segurança

O tempo de resposta de um dispositivo IoT responsável pelo fechamento de uma válvula de segurança de uma caldeira em uma aplicação industrial deve ser extremamente curto, especialmente se a falha representar risco às vidas humanas.

Para se atingir tempos de resposta determinísticos, o(a) arquiteto(a) deve utilizar sistemas operacionais de tempo real (RTOS), tomadas de decisão locais (sem a necessidade de interação síncrona com a nuvem), além de outros mecanismos que suportem este atributo de qualidade.

Figura 7.1: Exemplo de latência no qual uma máquina detecta uma situação de risco e reage imediatamente.

Exemplo de uso – Latência no processamento de mensagens

Um dos maiores desafios da arquitetura de sistemas é a capacidade de receber e processar, dentro do tempo esperado, grandes volumes de dados enviados por milhares de dispositivos. A adoção de serviços na nuvem possibilita o processamento desses grandes volumes de dados de forma virtualmente ilimitada.

A proximidade entre os serviços na nuvem dos dispositivos IoT é também um fator crítico para minimizar os tempos de resposta das transações. Provedores de nuvem permitem a criação de recursos em diversas localidades do globo, de modo que o dispositivo possa sempre se conectar ao servidor mais próximo ou àquele com o menor tempo de resposta.

7.3 Disponibilidade

Segundo o dicionário, disponibilidade é a qualidade de algo estar disponível para ser utilizado ou obtido. Essa propriedade também se aplica a um software, que deve estar pronto e disponível para atender às requisições dos usuários. A disponibilidade pode ser afetada por erros, falhas no hardware, ataques maliciosos ou sobrecargas nos recursos utilizados pela aplicação.

A disponibilidade é medida pelo tempo total em que o sistema esteve ativo e funcionando e pode ser definida pela função a seguir.

```
[1 - (tempo fora de operação/ tempo total)] x 100%
```

Como exemplo, podem-se citar os serviços na nuvem que objetivam otimizar sua disponibilidade a fim de atingir 99.999% (5-noves), o que representa indisponibilidade de 5 minutos por ano. Outros exemplos comuns são de disponibilidade de 99.99% (4-noves), ou indisponibilidade de 52 minutos por ano. Quanto maiores os requisitos de disponibilidade, maiores devem ser os investimentos

para garantir redundância (redes, armazenamento, processamento etc.), segurança e escalabilidade.

O Acordo de Nível de Serviço (SLA, do inglês *Service-Level Agreement*) é o compromisso assumido por um prestador de serviços. Esse acordo define a qualidade, as responsabilidades e a disponibilidade a serem cumpridas. O compromisso dos provedores de nuvem inclui tanto o funcionamento quanto a conectividade dos seus serviços. Podemos tomar como exemplo o serviço de armazenamento de dados provido pelo Microsoft Azure, cujo atual SLA é de 99.99% para leitura dos dados e de 99.9% para escrita (MICROSOFT, 2022e).

A tabela a seguir apresenta os níveis de disponibilidade de serviços e seus respectivos períodos de indisponibilidade tolerados por ano.

Disponibilidade	**Indisponibilidade por ano**
99.999%	5 minutos
99.99%	52 minutos
99.9%	526 minutos (8.77 horas)
99%	5256 minutos (87.6 horas)

Cabe ao arquiteto de IoT analisar o SLA de todos os serviços utilizados e identificar se eles se adequam à disponibilidade esperada para a solução.

Disponibilidade não se aplica apenas a serviços, mas também a dispositivos utilizados em campo: falhas no código, no hardware, na rede ou na sua fonte de energia não são acontecimentos raros. A disponibilidade de uma solução IoT também está associada à sua robustez, ou à sua capacidade de se recuperar de falhas. Existe uma relação direta entre robustez e disponibilidade: quanto mais robusta a solução, maior será sua disponibilidade.

Exemplo de uso – Tempo de vida da bateria do dispositivo

Para casos em que os dispositivos são alimentados por baterias, é necessário que o time de manutenção tenha acesso à informação do percentual restante da bateria para se antecipar ao fim da vida dela. Uma solução pode ser enviar essa informação (da condição da bateria) juntamente com a telemetria, de forma que será possível monitorar o estado da bateria de cada dispositivo remotamente. O dispositivo também pode identificar um cenário crítico de bateria e diminuir a frequência de conexões e do envio de telemetria, ou enviar uma mensagem específica notificando o sistema central de que se tornará indisponível até que sua bateria seja recarregada ou substituída.

Exemplo de uso – Erros no código do dispositivo

Erros podem tornar o dispositivo indisponível em cenários como *deadlocks*, estouro de memória, ou falhas no código. O dispositivo deve ser capaz de detectar que essa situação está ocorrendo e executar passos para restabelecer o seu estado de funcionamento esperado. Uma possível solução para esses casos é garantir que o dispositivo tenha uma forma de realizar um *auto-reset* sempre que necessário. Este mecanismo é conhecido como WDT (*watch-dog timer*), um temporizador programado para reiniciar o dispositivo quando o código não se comunicar com o WDT em um intervalo de tempo determinado.

Por exemplo, o desenvolvedor do dispositivo identifica a função/método que consome o maior tempo de processamento – digamos, 15 segundos – e configura o WDT para reiniciar o dispositivo se não receber uma comunicação do código a cada 60 segundos.

Exemplo de uso – Falhas na rede

Incidentes de conectividade são comuns e a pessoa arquiteta de IoT deve estar preparada para tomar decisões sobre como a solução deve se comportar nesses casos. O dispositivo deve ser capaz de detectar que uma falha de rede está ocorrendo, decidir o que fazer e retornar ao seu estado original quando a conectividade for restabelecida.

As equipes de manutenção devem ter acesso a ferramentas que identifiquem dispositivos que não se comunicaram com a nuvem por um determinado período de tempo – eles podem ter sido roubados ou danificados permanentemente.

Alguns cenários que ilustram casos de falha de rede:

- O dispositivo detecta uma falha na conexão e tenta reenviar os dados em intervalos de tempo aleatórios e incrementais (*backoff* com *jitter*);
- O dispositivo, mesmo sem conexão de rede, deve reter os dados para serem enviados quando a conexão voltar a funcionar. Nesse caso, a quantidade de memória disponível é importante para definir quanto tempo o dispositivo pode ficar sem rede sem que haja perda de dados. Também é importante definir se a memória onde esses dados são armazenados é permanente ou volátil: tudo depende da criticidade do dispositivo e da informação que eles coletam.
- Necessidade de redundância de rede para minimizar interrupções na conexão – o dispositivo com conexão wi-fi pode alternativamente usar uma rede celular para se conectar; um dispositivo celular pode usar rádio de satélite, e assim por diante. Nesse caso, dado que o custo da redundância de rede tende a ser maior, o dispositivo deve ser capaz de detectar que a rede primária está novamente disponível para reverter a troca.

7.4 Usabilidade

Usabilidade é a capacidade de um sistema prover uma condição de uso em que os usuários executam tarefas de forma segura, eficiente e prazerosa. Características da usabilidade incluem a facilidade de aprendizado, a eficiência no uso da aplicação, a minimização de erros, a adaptação dos sistemas às necessidades dos usuários e o aumento da confiança de que o sistema realizará de forma correta as ações desejadas.

Ao considerarmos uma solução IoT, é preciso identificar como a usabilidade de seus dispositivos pode contribuir para a experiência do usuário. A seguir, são apresentados alguns dos aspectos que devem ser considerados:

- **Interfaces de dispositivos IoT** – nem todos os dispositivos IoT possuem interface visual (tela/display). Algumas são extremamente simples contando apenas com um botão ou microfone; outras são fortemente baseadas em aplicativos para celular de onde o dispositivo é efetivamente controlado.
- **Tipos de interface do dispositivo** – dispositivos IoT contam com uma grande variedade de interfaces possíveis como botões, *buzzers* para alertas sonoros, LEDs de diferentes cores, telas LCD / matriciais / OLED, motores de vibração para feedback, entre outros. Esses tipos de interfaces podem ser usados em combinação uns com os outros para aumentar a usabilidade e o alcance de usuários que podem utilizar o dispositivo, incluindo pessoas com deficiência visual, auditiva ou dificuldades motoras. Cabe ao(à) arquiteto(a) de IoT identificar as necessidades dos usuários da aplicação para desenvolver uma interface que atenda a essas necessidades.
- **Confirmação e ação** – a interface de usuário do dispositivo deve permitir o cancelamento de operações e que este ocorra dentro do tempo esperado. Em cenários críticos, o tempo de resposta de um comando de parada deve ser concluído dentro de um tempo determinado (seguindo normas de segurança na indústria). Especificamente, em ambientes industriais, interfaces

de usuários em máquinas são conhecidas como HMI (*Human Machine Interface*).

- **Interação com aplicativos móveis** – a combinação entre dispositivos IoT e interfaces em aplicativos para smartphones são comuns e, nesse caso, a familiaridade dos usuários com smartphones facilita o trabalho de desenvolver uma interface simples e intuitiva.

7.5 Segurança

De acordo com Bass *et al.* (2021), a segurança é uma medida da capacidade do sistema de proteger dados e informações de acesso não autorizado, enquanto mantém o acesso a usuários e sistemas autorizados. Um ataque, ou seja, uma ação realizada contra um sistema computacional com a intenção de causar danos, pode assumir várias formas: acesso de dados ou serviços, tentativa de modificar dados ou a intenção de negar serviços a usuários legítimos.

Com relação à definição dos requisitos de segurança de uma solução, são de responsabilidade do(a) arquiteto(a) de IoT as seguintes atividades:

- Mapear a superfície de ataque de cada elemento da solução, desde o dispositivo, passando pela rede até os elementos da solução na nuvem;
- Definir o nível de segurança de acesso físico e lógico dos dispositivos;
- Definir como os dispositivos vão proteger os canais de comunicação com a nuvem e entre dispositivos (incluindo *gateways*);
- Definir táticas de proteção, detecção, reação e recuperação dos dados em caso de roubo de dispositivos ou tentativa de ataque

à solução:

- Proteção – adotar medidas que restrinjam o acesso à informação contida nos elementos da solução;
- Detecção – identificar a possível atuação de um usuário mal-intencionado a partir de anomalias na execução de um dos elementos da solução;
- Reação – uma vez detectada uma invasão, a reação pode variar entre reportar a tentativa e continuar a operação, até mesmo invalidar credenciais e bloquear o acesso do dispositivo comprometido;
- Recuperação – recuperar a integridade dos elementos da solução para que ela retorne à sua operação normal.

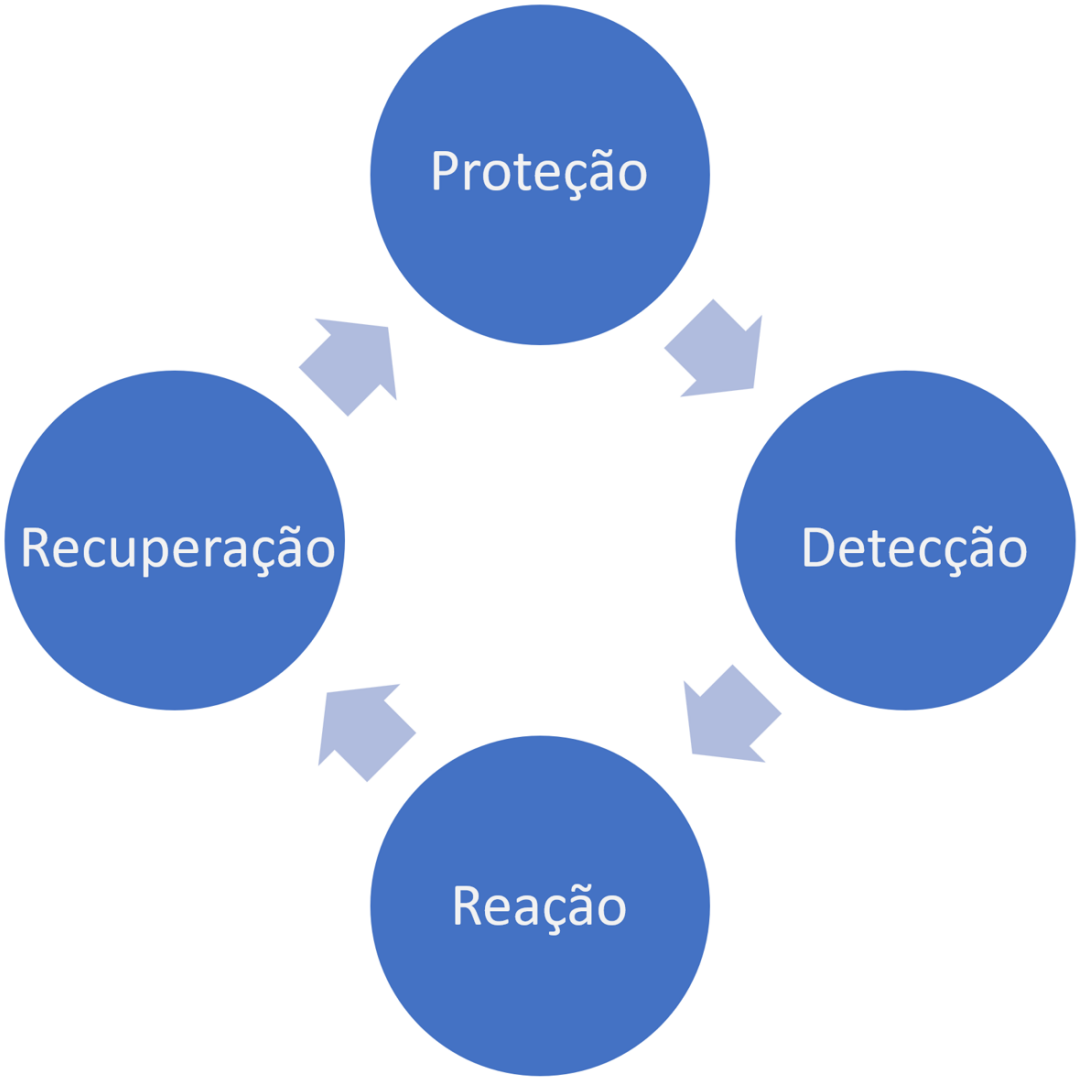

Figura 7.2: Ciclo de táticas de segurança.

Uma frase comum em artigos de IoT é uma provocação que diz: "*A letra **S** em IoT refere-se à Segurança*". Essa é uma crítica clara ao fato de que a segurança tende a ser deixada de lado em soluções de IoT.

Do ponto de vista de segurança, o ponto mais vulnerável de uma solução IoT são os seus dispositivos. Ao contrário dos servidores da nuvem, que permanecem dentro dos *data centers*, atrás de barreiras físicas e lógicas que impedem seu acesso por pessoas não autorizadas, os dispositivos IoT, por força da necessidade de negócio, estão em campo, o que facilita ações como interceptação

de comunicação, exposição dos componentes internos e até mesmo o roubo do dispositivo inteiro.

A proteção de aplicações e dispositivos em campo pode ser altamente complexa, como ilustrado no artigo *The Seven Properties of Highly Secured Devices* (HUNT *et al.*, 2017), que mostra como a segurança de produtos como o Azure Sphere foram criados. A segurança dos dispositivos, a complexidade de implementação, o custo de aquisição e a manutenção podem variar imensamente e a recomendação geral é a de adicionar o nível de segurança necessário de acordo com cenário de negócio.

No quesito segurança de dispositivos, a própria conexão com serviços na nuvem já define o patamar mínimo aceitável – por exemplo, ao exigir a criptografia baseada em TLS (*Transport Layer Security*), que garante que toda comunicação entre o dispositivo e o *endpoint* da nuvem será criptografada.

Uma vez que a criptografia do *endpoint* está estabelecida, o próximo passo é verificar a identidade do dispositivo, o que pode ocorrer de diversos modos, sendo os mais comuns:

- **Token** – geração de um *token* baseado em uma chave secreta conhecida pelo dispositivo e pelo serviço na nuvem. Normalmente esse *token* tem validade muito curta (30 minutos) e precisa ser gerado novamente quando o *token* anterior expira. Esse modelo é baseado em criptografia simétrica, onde ambas as partes compartilham a mesma chave secreta.
- **Certificados X.509 autoassinados** – neste caso, tanto o certificado raiz, certificados intermediários e certificados de dispositivos são gerados com base em um *engine* como o OpenSSL, sem o intermédio de uma entidade certificadora. Os dispositivos passam a usar certificados digitais para se identificarem junto ao serviço na nuvem.
- **Certificados X.509 assinados por entidade certificadora** – certificados X.509 são adquiridos de entidades certificadoras

como CertiSign e VeriSign. Neste caso, não só os dispositivos possuem um certificado único para verificar sua identidade, mas este certificado pertence a uma cadeia que é assinada e validada por uma entidade certificadora.

Independente do método de conexão e validação de identidade, existe ainda a necessidade de se manterem protegidos os segredos (certificados, chaves primárias e secundárias) utilizados pelos dispositivos, caso um destes seja comprometido. Uma das formas de se proteger segredos no dispositivo é por meio do uso de um HSM (*Hardware Security Module*). Esse módulo provê segurança baseada em hardware com um chip otimizado para realizar funções criptográficas e que pode armazenar certificados e suas chaves de forma segura. Um exemplo de HSM é o TPM (*Trusted Platform Module*), muito utilizado em servidores, desktops e laptops.

7.6 Escalabilidade

Escalabilidade é a habilidade de um sistema em responder ao aumento da carga de trabalho adicionando novos recursos computacionais para garantir o desempenho esperado por seus clientes. Soluções IoT precisam adaptar-se ao volume de dados, requisições trafegadas e quantidade de dispositivos, sendo essa carga de trabalho constante ou com aumentos esporádicos.

O conceito de escalabilidade é amplamente utilizado em serviços oferecidos por provedores de nuvem e pode ser executado de duas maneiras:

- Horizontal (*scale-out*) – adição ou remoção de recursos a unidades lógicas, como a adição ou remoção de um nó a um *cluster* de servidores;
- Vertical (*scale-up*) – refere-se à adição ou remoção de recursos a uma unidade física, como a adição ou remoção de memória

de um servidor.

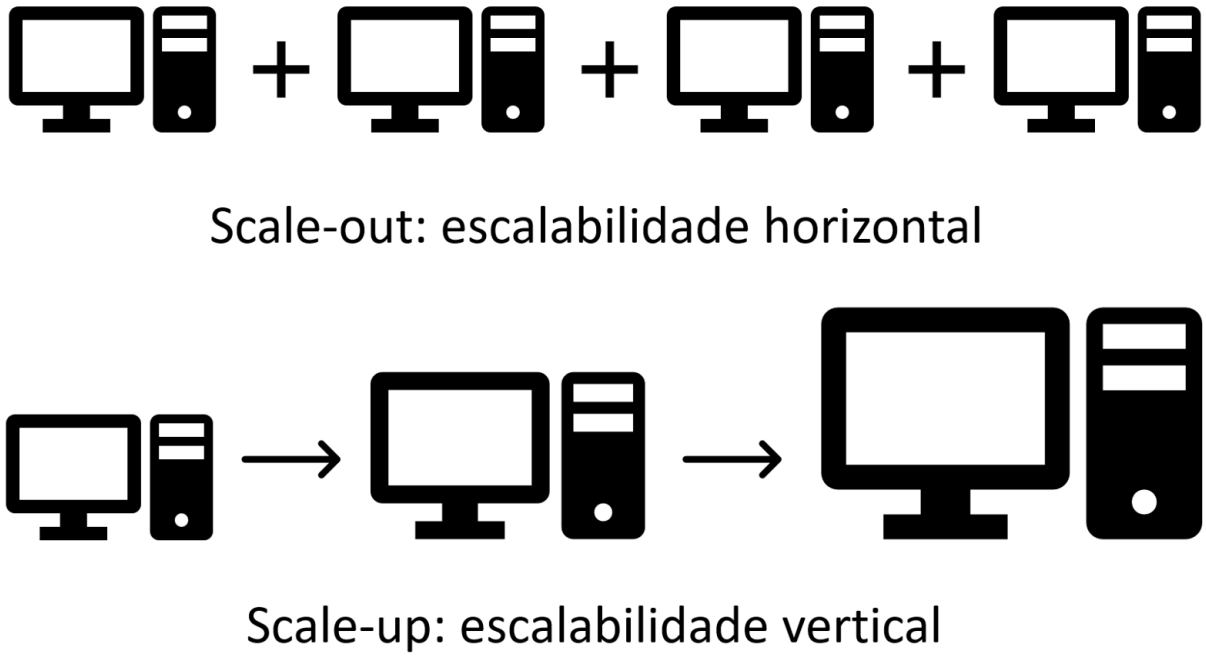

Figura 7.3: Representação dos modelos de escalabilidade horizontal e vertical.

Apesar de ser amplamente utilizado na nuvem, escalabilidade é um critério de qualidade que deve ser considerado em toda a solução. É preciso garantir que campo e conectividade também vão escalar conforme o aumento da demanda. Para tanto, alguns aspectos precisam ser considerados:

Provisionamento de dispositivos – arquitetos(as) de IoT precisam entender se suas soluções são capazes de conectar dispositivos de forma eficiente e segura, além de permitir que esses dispositivos possam ser provisionados. Precisam também prever cenários que acarretam no aumento de provisionamentos simultâneos de dispositivos, e que possam impactar a adesão de novos clientes.

Over-the-Air Update – a adoção de um modelo automatizado e seguro para atualização de dispositivos é vital, assim como é necessário conhecer os limites suportados pelas ferramentas de OTA, incluindo, por exemplo, o número máximo de atualizações concorrentes e o tempo gasto na atualização dos dispositivos. Ao

conhecer os limites suportados, estratégias podem ser adotadas para garantir o melhor funcionamento da solução e minimizar o tempo gasto na atualização de dispositivos. Tais estratégias podem ser criadas para cenários como:

- Atualizações rotineiras de dispositivos;
- Atualizações críticas (*hotfix* ou segurança) de dispositivos;
- Atualizações por região e/ou grupos de dispositivos;
- Atualizações por período de tempo.

Limite de cobertura e conexões em campo – escalabilidade de dispositivos corresponde ao número de dispositivos que uma solução IoT é capaz de suportar com desempenho aceitável. É preciso conhecer os limites das tecnologias e recursos utilizados em campo para garantir que a adição de novos dispositivos seja feita dentro dos limites atualmente suportados pela infraestrutura. Como exemplo, podemos citar *gateways* Z-Wave que suportam até 232 dispositivos conectados ao mesmo tempo, ou até mesmo roteadores wi-fi, que, em sua maioria, suportam até 250 dispositivos simultaneamente.

Segurança – uma solução escalável permite que mais dispositivos se conectam a ela; da mesma forma, o aumento no número de dispositivos conectados cria maiores riscos de segurança. Isso acontece porque, a cada novo dispositivo, a superfície de ataque da solução é ampliada. Manter altos níveis de segurança entre dispositivos e redes, ao passo em que uma solução escala, é desafiador. Segurança é um atributo de qualidade crítico para o sucesso de uma solução IoT, e quanto maior esta solução for, maiores deverão ser as precauções tomadas.

Limitações de espaço e energia – o planejamento da solução IoT também deve considerar o espaço físico e o fornecimento de energia necessários para a adição de novos dispositivos à solução, caso haja necessidade. Ao prever e endereçar essas demandas antes da implantação da solução, o(a) arquiteto(a) de IoT pode estar

mitigando possíveis impeditivos que atrasariam o crescimento da solução para atender novos clientes.

Para que uma solução seja escalável, não é suficiente atender apenas a aumentos esporádicos no volume e na carga de trabalho atuais; é preciso estar preparado para o aumento que a adição de novos clientes trará ao longo do tempo. A disciplina de monitorar a demanda da aplicação é fundamental para compreender se o seu plano atual de escalabilidade é o ideal e se ações preventivas são necessárias.

7.7 Manutenibilidade

Refere-se à habilidade do sistema de suportar mudanças. Tais mudanças podem ser necessárias em decorrência de erros que precisam ser corrigidos, de melhorias no código e na segurança ou de adaptações a novas regras de negócio ou legislação.

Manter dispositivos IoT atualizados não é uma tarefa trivial. Mesmo no caso dos dispositivos baseados em sistemas operacionais como Windows ou Linux, para os quais existem diversas alternativas de atualização e gerenciamento de pacotes e dependências, ainda existe a possibilidade de tais dispositivos estarem em locais remotos, fazendo com que o processo de atualização seja mais complexo.

Para a atualização de dispositivos, a pessoa arquiteta de IoT deve considerar:

- Notificação do dispositivo sobre a necessidade de realizar um update;
- Frequência com a qual o dispositivo deve verificar a existência de novas versões;
- Como o download dos pacotes será realizado;

- Garantir que a notificação de atualização, assim como os pacotes recebidos, são de origem conhecida;
- Definir melhor período para a atualização dos dispositivos a fim de minimizar impactos em suas tarefas normais;
- Método de validação da atualização após instalada;
- Definir estratégia de *rollback* da atualização;
- Laboratório de pré-produção com dispositivos reais para validar e assegurar a completude do processo de atualização.

No caso de dispositivos baseados em microcontroladores, nos quais a capacidade de processamento e memória são limitadas, pode ser necessário quebrar o pacote de atualização em partes menores para possibilitar sua instalação. Ainda assim, existem situações em que a atualização dos dispositivos pode não ser possível dadas limitações de hardware, como:

- **Limite de memória:** quando o pacote de atualização é maior que o espaço livre em memória, uma possível alternativa é a quebra do pacote de instalação em pedaços menores.

- **Limite de conexões simultâneas:** a maioria das soluções de OTA utiliza CDN (*Content Delivery Network*), que utiliza o protocolo HTTP(S), mas nem todos os dispositivos baseados em microcontroladores suportam múltiplas conexões simultâneas para o tráfego de telemetria e comandos via MQTT e o download dos pacotes via HTTP(S).

- **Limite de processamento:** atualizações de segurança que exijam maior capacidade de processamento do que a disponível no dispositivo. Por exemplo, a mudança do algoritmo de criptografia de certificados de SHA256 para SHA384 requer um significativo incremento da capacidade do processador para lidar com chaves maiores. O dispositivo pode não ser capaz de lidar com esse incremento e responder a requisições do servidor dentro do tempo esperado.

7.8 Eficiência no consumo de energia

Segundo Bass *et al.* (2021), com o domínio dos dispositivos móveis como a principal forma de computação para a maioria das pessoas e com a crescente adoção da Internet das Coisas (IoT), o consumo de energia pelos dispositivos tornou-se uma preocupação que os arquitetos devem endereçar, pois este é um atributo crítico para viabilizar soluções IoT.

Exceto em casos de desastre, as plataformas de nuvem normalmente não precisam se preocupar com a falta de energia devido aos mecanismos de redundância de energia, enquanto essa é uma preocupação diária para usuários de dispositivos móveis e dispositivos IoT.

O consumo de energia de um dispositivo influencia diretamente em outros atributos de qualidade, tais como desempenho, disponibilidade, segurança e funcionalidade. Em todos os casos, a eficiência de energia deve ser equilibrada com os demais atributos de qualidade, exigindo que os arquitetos de IoT considerem os *trade-offs* de suas decisões no consumo de energia dos dispositivos.

Dependendo do cenário de uso do dispositivo IoT - dado o seu tamanho, peso e portabilidade -, o tamanho de sua bateria pode ser restritivo. *Wearables* são um exemplo no qual o formato do dispositivo, alinhado ao tamanho da bateria, precisa funcionar com perfeição, pois um dispositivo muito pesado ou cuja bateria não o suporte ao longo do dia pode ser proibitivo.

Conclusão

Construir soluções escaláveis, seguras, manuteníveis e com bom desempenho não é trivial. Conhecer os requisitos da aplicação, alinhá-los às tecnologias disponíveis e identificar os *trade-offs* são passos essenciais para construção de soluções IoT.

É de responsabilidade da pessoa arquiteta garantir que os requisitos e atributos de qualidade sejam atendidos, acompanhando o andamento do projeto e mantendo a aderência com o design da solução.

Capítulo 8
Conclusão

Para você que chegou até este ponto do livro, vale ressaltar que todo o conhecimento adquirido até aqui é apenas o primeiro passo de uma longa jornada, não o último. A experiência na construção de soluções IoT faz parte de um processo incremental no qual, a cada projeto, novos conhecimentos são assimilados.

Ao longo desta publicação, foram apresentados os tipos de dispositivos IoT, suas características e possíveis usos. Também foram apresentados tipos de rede e protocolos que permitem a comunicação desses dispositivos com serviços na nuvem, viabilizando a criação de soluções IoT mais seguras, escaláveis, com maior desempenho e manutenibilidade.

Dentre os conceitos apresentados, vimos que não existe uma única prática que funcione para todos os cenários de IoT: cada uma é bastante particular, considerando os seus requisitos. É preciso que os profissionais envolvidos na implementação de soluções IoT continuem ampliando seu conhecimento e adicionando novas ferramentas em seu "cinto de utilidades”.

A IoT é uma área do conhecimento relativamente nova e em constante evolução, na qual se têm observado as seguintes tendências:

- Segurança – melhorias na área de segurança para garantir que o crescimento exponencial de dispositivos não seja um fator de risco para soluções;
- Extrair valor dos dados gerados – novas ferramentas para auxiliar na extração de valor a partir do grande volume de dados gerados pelos dispositivos de uma solução;
- Novas verticais – tem-se observado a especialização em novas verticais a partir de IoT, tais como IoMT (*Internet of Medical*

Things) e AIoT (*Artificial Intelligence of Things*).

É esperado que IoT se torne comum em mais setores ao longo dos próximos anos, graças à popularização do consumo de nuvem e ao desenvolvimento de tecnologias de comunicação móvel (tal como o 5G).

Eletrodomésticos inteligentes, sistemas de segurança conectados, smartphones e carros cada vez mais conectados e cheios de tecnologias já fazem parte do nosso dia a dia e têm melhorado a experiência de seus usuários como um todo. Isso nos leva a acreditar que, em breve, a IoT se tornará *commodity*, e não mais um diferenciador de destaque de um produto sobre os demais, o que tornará cada vez mais popular o uso da Computação Ubíqua e o mundo, mais conectado.

Capítulo 9
Referências

ABOUT Z-Wave technology. *Z-Wave Alliance*, 2022. Disponível em: https://z-wavealliance.org/about_z-wave_technology. Acesso em 19 set. 2022.

ANDRADE, R. M. *Um roteiro para o ensino de qualidade de arquitetura de software guiado por requisitos não funcionais*. Tese (Doutorado em Sistemas Digitais) - Escola Politécnica, Universidade de São Paulo, São Paulo, 2015. Disponível em: https://teses.usp.br/teses/disponiveis/3/3141/tde-19072016-121434/pt-br.php. Acesso em 19 set. 2022.

ARELLANO, J. Bluetooth vs. Wi-Fi for IoT: Which is Better? *Very*, 2019. Disponível em https://www.verypossible.com/insights/bluetooth-vs.-wi-fi-for-iot-which-is-better. Acesso em setembro de 2022.

ATmega328P - 8-bit AVR Microcontroller with 32K Bytes In-System Programmable Flash DATASHEET. *Atmel Corporation*, 2015. Disponível em: http://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P_Datasheet.pdf. Acesso em 19 set. 2022.

BASS, L.; CLEMENTS, P.; KAZMAN, R. *Software architecture in practice*. 3. ed. Boston: Addison-Wesley, 2012.

BLUETOOTH Wireless Technology. *Bluetooth SIG*, 2022. Disponível em: https://www.bluetooth.com/learn-about-bluetooth/tech-overview/. Acesso em 19 set. 2022.

CHUMAK , I. LoRa: Enabling a Cost-Effective, Low-Energy IoT Infrastructure. *PLVision*. 2018. Disponível em: https://plvision.eu/rd-lab/blog/iot/lora-enabling-cost-effective-low-energy-iot-infrastructure. Acesso em 19 set. 2022.

CLEMENTS, P.; BACHMANN, F.; BASS, L.; GARLAN, D.; IVERS, J.; LITTLE, R.; MERSON, P.; NORD, R.; STAFFORD, J. *Documenting software architectures*: Views and beyond. SEI Series in Software Engineering. 2ª ed. Addison-Wesley Educational, 2010.

COULOURIS, G. F.; DOLLIMORE, J.; KINDBERG, T.; BLAIR, G. *Distributed Systems*: Concepts and Design. 5ª ed. Addison Wesley, 2011.

FALUDI, R. How Do IoT Devices Communicate?. *DIGI*, 2021. Disponível em: https://www.digi.com/blog/post/how-do-iot-devices-communicate. Acesso em 19 set. 2022.

FAZENDA vertical e sua importância para o futuro da agricultura, A. *Tecnologia no Campo*, 2020. Disponível em: https://tecnologianocampo.com.br/fazenda-vertical. Acesso em 19 set. 2022.

FRENZEL, L. What's The Difference Between ZigBee And Z-Wave? *Electronic Design*, 2012. Disponível em: https://www.electronicdesign.com/technologies/communications/article/21796052/whats-the-difference-between-zigbee-and-zwave. Acesso em 19 set. 2022.

HUNT, G. C.; LETEY, G.; NIGHTINGALE, E. *The Seven Properties of Highly Secure Devices*. Microsoft Azure Sphere Team. 2. ed. 2020. Disponível em: https://www.microsoft.com/en-us/research/uploads/prod/2020/11/Seven-Properties-of-Highly-Secured-Devices-2nd-Edition-R1.pdf. Acesso em 19 set. 2022.

IEEE 802.15.4 Standard: a tutorial/primer. *Electronics Notes*, 2021. Disponível em: https://www.electronics-notes.com/articles/connectivity/ieee-802-15-4-wireless/basics-tutorial-primer.php. Acesso em 19 set. 2022.

ISO/IEC 25010: Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — System and software quality models. *International Organization for Standardization*. ISO/IEC, 2011. Disponível em: https://www.iso.org/obp/ui/#iso:std:iso-iec:25010:ed-1:v1:en. Acesso em 19 set. 2022.

KHVOYNITSKAYA, S. The IoT history and future. *Itransition*, 2019. Disponível em: https://www.itransition.com/blog/iot-history. Acesso em 19 set. 2022.

LEA, P. *IoT and Edge Computing for Architects*: Implementing edge and IoT systems from sensors to clouds with communication systems, analytics, and security. 2. ed. United Kingdom: Packt Publishing, 2020.

PECOSQUI, E. Risco de Explosão em Silos. *LinkedIn*, 2019. Disponível em: https://www.linkedin.com/pulse/riscos-de-explos%C3%A3o-em-silos-edmilson-pecosqui. Acesso em 19 set. 2022.

PEREIRA, C. *Profibus*. Universidade Federal do Rio Grande do Sul. Departamento de Engenharia Elétrica. 2002. Disponível em:

http://www.ece.ufrgs.br/~cpereira/CursoBarramentos_Novo/profibus_overview.pdf. Acesso em 19 set. 2022.

QUE é IaaS? Infraestrutura como serviço, O. *Microsoft*, c2022a. Disponível em: https://azure.microsoft.com/pt-br/overview/what-is-iaas. Acesso em 19 set. 2022.

QUE é PaaS? Plataforma como serviço, O. *Microsoft*, c2022b. Disponível em: https://azure.microsoft.com/pt-br/overview/what-is-paas. Acesso em 19 set. 2022.

QUE é SaaS? Software como serviço, O. *Microsoft* c2022c. Disponível em: https://azure.microsoft.com/pt-br/overview/what-is-saas. Acesso em 19 set. 2022.

SECURE, SUSTAINABLE SMART cities and the IoT. *Thales*, 2022. Disponível em: https://www.thalesgroup.com/en/markets/digital-identity-and-security/iot/inspired/smart-cities. Acesso em 19 set. 2022.

SLA for Storage Accounts. *Microsoft* c2022e. Disponível em: https://azure.microsoft.com/en-us/support/legal/sla/storage/v1_5. Acesso em 19 set. 2022.

UNDERSTAND and use device twins in IoT Hub. *Microsoft* c2022d. Disponível em: https://docs.microsoft.com/en-us/azure/iot-hub/iot-hub-devguide-device-twins. Acesso em 19 set. 2022.

UNDERSTANDING Bluetooth Range. *Bluetooth SIG*, 2022. Disponível em: https://www.bluetooth.com/learn-about-bluetooth/key-attributes/range/. Acesso em 19 set. 2022.

USE FALL DETECTION with Apple Watch. *Apple Inc.*, c2022. Disponível em: https://support.apple.com/en-us/HT208944. Acesso em 19 set. 2022.

VOITOVYCH, B. What Makes LoRaWAN Mesh a Great Smart Home and City Solution? *Lemberg Solutions*, 2019. Disponível em: https://lembergsolutions.com/blog/what-makes-lorawan-mesh-great-smart-home-and-city-solution. Acesso em 19 set. 2022.

WAQAS. Hackers attack Casino's fish tank thermometer to obtain sensitive data. *Hackread*, 2018. Disponível em: https://www.hackread.com/hackers-casinos-fish-tank-smart-thermometer-hack/. Acesso em 19 set. 2022.

WHAT is LoRaWan specification? *LoRa Alliance*, 2022. Disponível em: https://lora-alliance.org/about-lorawan. Acesso em 19 set. 2022.

ZIGBEE The Full-Stack Solution for All Smart Devices. *Connectivity Standards Alliance*. c2022. Disponível em: https://csa-iot.org/all-solutions/zigbee/. Acesso em 19 set. 2022.

5 WAYS smartwatches could improve your health. *Northwest Primary Care Blog*, 2022. Disponível em: https://www.nwpc.com/how-smartwatches-could-improve-your-health. Acesso em 19 set. 2022.